A Comprehensive Exploration of Models and Techniques for Sentiment Analysis in Amazon Product Reviews

Katya Trufanova

***Abstract*—This project explores the application of various machine learning models and techniques to perform sentiment analysis on Amazon product reviews, classifying them into positive, negative, or neutral categories. Techniques such as hyperparameter tuning, ensemble learning, pre-trained GloVe embeddings, Local Interpretable Model-agnostic Explanations (LIME), Synthetic Minority Over-sampling Technique (SMOTE), dimensionality reduction, feature extraction, engineering and selection were employed. Experiment results indicate that biLSTM and DistilBERT perform optimally compared to simpler models like Naive Bayes, Logistic Regression, and SVM. Interestingly, the techniques used in this study led to either decresead or equivalent performance, suggesting that a more basic approach might be the most appropriate in this context.**

I. Introduction

Sentiment analysis, an integral part of Natural Language Processing (NLP), has emerged as a critical tool to extract and analyze subjective information from textual data. This project report presents a comprehensive exploration of the application of sentiment analysis on Amazon product reviews. The aim was to classify reviews into one of three sentiment categories: positive, negative, or neutral. This task was undertaken in response to the increasing growth of online shopping and the consequent surge in user-generated content, which poses challenges for manual review and analysis.

The project was built upon the premise that sentiment analysis could provide valuable insights into customer satisfaction and product performance, going beyond the rudimentary understanding provided by star ratings. The wide applicability and potential impact of sentiment analysis in various fields, including political science, finance, and social sciences, was another compelling factor underlying the choice of this task.

In a deviation from the binary classification (positive versus negative) commonly encountered in research, this project approached sentiment analysis as a ternary or multi-class classification problem. This decision was motivated by the desire to enhance the granularity of the analysis, thereby enabling a more detailed insight into the sentiment of the reviews.

This paper provides a detailed account of the dataset description, models, and methods used, followed by an exhaustive presentation of the experimental setting, which includes the experimental design, data preprocessing, synthetic minority oversampling, and local interpretable model-agnostic explanations. The experimental results and an extensive analysis of these results, focusing on performance analysis and discussion, are then presented. The paper concludes with recommendations for future work and a summary of the project's findings.

II. Related work

This section delineates the body of literature that has been consulted and drawn upon in the course of this project. The reviewed literature spans various domains, including sentiment analysis techniques, machine learning models, and feature engineering strategies.

A. Sentiment Analysis Techniques

A comprehensive survey of sentiment analysis techniques applied to product review analysis was conducted by Ezhilarasan et al. [1]. The authors provided an in-depth overview of various methods, ranging from rule-based and lexicon-based methods to machine learning-based techniques. The paper also discussed the inherent challenges in sentiment analysis, including handling sarcasm, irony, and subjectivity, which are prevalent in product reviews. The notable conclusion was the superior efficiency of machine learning-based strategies, particularly support vector machines and neural networks, in sentiment analysis for product reviews. Furthermore, they emphasized the need for continuous research to develop more accurate models for sentiment analysis in product reviews. This survey formed the basis for the selection of simpler models, including Naive Bayes, Logistic Regression, and SVM, in the current project.

B. Machine Learning Models for Sentiment Analysis

The use of machine learning models for sentiment analysis has been extensively explored in literature. Ni and Cao [2], for instance, proposed a model that combines GloVe and LSTM-GRU for sentiment analysis. The authors used the GloVe model to generate word embeddings and the LSTM-GRU model for sentiment classification. This model was trained and tested on the IMDB movie review dataset and achieved an impressive accuracy of 89.5%. The model was benchmarked against other state-of-the-art models, such as the Bag-of-Words model, the Word2Vec model, and the LSTM model, and it outperformed all of them in terms of accuracy. This paper provided the foundational work for the use of biLSTM with GloVe embeddings in this project.

Moreover, the application of biLSTM for sentiment analysis without GloVe embeddings as a baseline experiment was

supported by several studies, including [3], [4], and [5], all of which reported high performance of the bi-LSTM model in sentiment analysis tasks.

A novel approach to sentiment analysis using DistilBERT was proposed by Almuayqil et al. [6]. The authors combined random majority under-sampling (RMU) with machine learning models to address the issue of class imbalance in sentiment analysis datasets. They reported an accuracy of 87.6% using DistilBERT, which was significantly higher than other models tested. Moreover, their approach reduced the time complexity of sentiment analysis, making it more efficient. This work was instrumental in the selection of the DistilBERT model for sentiment analysis in this project.

C. Feature Engineering and Selection

The process of feature engineering and selection was guided by the book "Feature Engineering and Selection: A Practical Approach for Predictive Models" by Kuhn and Johnson [7]. This book offers a comprehensive guide on transforming raw data into meaningful features that can enhance the performance of predictive models. The authors covered a wide range of topics, from data preprocessing and feature extraction to feature construction and feature selection. The book also emphasized the critical role of domain knowledge and provided insights into various feature engineering methods. The process of feature selection, which entails identifying the most relevant and informative features for a predictive model, was also thoroughly discussed. This book served as a valuable resource for the feature engineering and selection in this project.

III. Task Description

The project's primary target was to perform sentiment analysis on Amazon product reviews. The reviews were classified into one of three distinct sentiment categories: 'positive', 'negative', or 'neutral'. Sentiment analysis, a widely used aspect of Natural Language Processing (NLP), facilitates the extraction of subjective information from textual data. In the given context, it is particularly beneficial for an in-depth examination of customer feedback regarding products.

The choice of this task was motivated by several considerations. Primarily, the globalization of online shopping has caused a surge in user-generated content, such as product reviews on platforms like Amazon. However, the sheer volume of this data brings challenges in manual analysis. The application of sentiment analysis to such data can yield valuable insights into customer satisfaction and product performance, insights that are not straightforwardly discernible from star ratings alone.

Additionally, sentiment analysis holds a certain fascination due to its inherent intricacy. Understanding and interpreting human emotions is not a simple feat. Emotions are largely subjective and can be influenced by a variety of factors, including cultural, contextual, and personal. The task of training machines to accurately identify these sentiments is a demanding but intriguing endeavor.

Sentiment analysis is also critically important in many practical situations. For businesses, it provides an opportunity to assess customer opinions and feedback on a large scale, invaluable for product development and improvement. For customers, an aggregate sentiment score can offer a more nuanced view of a product's reception than individual reviews or average star ratings.

Given these factors, sentiment analysis of Amazon product reviews was deemed an appropriate and significant task. It encapsulates the intricate blend of NLP, machine learning, and data analysis, offering a multifaceted challenge that can lead to rich insights.

Sentiment analysis's implications extend beyond the commercial realm. It is increasingly employed in political science for analyzing social media posts, in finance for predicting stock market trends based on news sentiment, and in social sciences for assessing public opinion on various issues. This wide applicability further emphasizes its relevance and potential impact.

For the task of sentiment analysis, models need to be developed that can accurately classify text according to the sentiment it expresses. This task involves using various NLP techniques and machine learning algorithms, which are detailed in the following sections of this report. The effectiveness of these models was assessed using a dataset of Amazon product reviews, the description and analysis of which are also provided in the subsequent sections.

In this project, sentiment analysis was approached as a ternary, or multi-class, classification problem. This decision was made to deviate from the commonly encountered binary classification research (positive/negative) in favor of a slightly more intricate task that improves the granularity of the analysis. This finer granularity allows for more detailed insights into the sentiment of the reviews.

For instance, a review classified as neutral could contain valuable feedback that could contribute to product improvement. By incorporating a 'neutral' category, it might be possible to discern and highlight such nuanced feedback that might otherwise be lost in a binary classification system. This approach allows for a more comprehensive understanding of customer opinions and makes room for more precise product enhancements based on this feedback.

IV. Dataset Description

The dataset employed in this project has been sourced from Kaggle, specifically the "Amazon Reviews: Unlocked Mobile Phones" dataset [8]. This dataset, which was published by PromptCloud, a web scraping company, in 2016, encompasses more than 400,000 reviews that are part of Amazon's "Unlocked Mobile Phone" category.

The dataset is structured into six distinct columns:

1) Brand Name: This column represents the name of the manufacturer or company that produced the mobile phone. For instance, one entry might be "Nokia".
2) Product Name: This field displays the specific model of the mobile phone, such as "Nokia Asha 302".
3) Price: This column indicates the cost associated with each mobile phone.
4) Rating: This presents the star rating given to the product by the customer on a scale from 1 to 5.

5) Reviews: This section contains the text of the reviews provided by users regarding each product.
6) Review Votes: This column lists the number of consumers who found the review helpful, providing information about the credibility of the review.

The dataset was extracted from Amazon.com, a rich source of product reviews. In the modern consumer landscape, consumers increasingly seek insights from fellow consumers, alongside the information provided by the seller. As such, reviews and ratings have become integral to customers' decision-making process. Amazon's review system is considered to be transparent and accessible, allowing consumers to make informed purchase decisions.

The data was procured in December 2016 through the use of specialized web crawlers for data extraction services. With over 400,000 reviews, the dataset provides a comprehensive overview of nearly 4,400 distinct unlocked mobile phone models.

The majority of the reviewers assigned either 4-star or 3-star ratings, with a significantly smaller percentage awarding a 1-star rating. The mean value of all ratings was calculated to be 3.62. When considering the length of reviews, it was found that most contained fewer than 300 characters, with the mean length being roughly 230 characters. This suggests that reviewers generally prefer writing concise reviews, typically spanning one to two sentences.

V. Models and Methods

In this project, a comprehensive approach was adopted, deploying a variety of machine learning models and techniques to ensure a thorough exploration of the task at hand. The models and techniques selected were based on the explored related work, their relevance to the task, their success in similar kinds of tasks, and the opportunity they provide to perform a broad exploration of machine learning concepts and applications.

A. Models

The following models were used in this project:

1) Naive Bayes
2) Logistic Regression
3) Support Vector Machine (SVM)
4) Bi-directional Long Short-Term Memory (biLSTM)
5) DistilBERT

Naive Bayes is a probabilistic classifier based on Bayes' theorem with an assumption of independence between features. The simplicity of this model and its ability to handle large feature spaces make it a good starting point for text classification tasks. Its assumptions, while naive, often hold reasonably well for tasks like sentiment analysis, where the presence or absence of certain words (features) can independently contribute to the overall sentiment of a review.

Logistic Regression is a statistical model that models the probability of the input belonging to a particular class using a logistic function. For the task of sentiment analysis, it was chosen due to its efficiency. As a matter of fact, Logistic Regression is a linear model, which makes it computationally efficient and scalable to large datasets.

Support Vector Machine (SVM) is a powerful machine learning algorithm that is commonly used for classification tasks. SVM works by finding the optimal hyperplane that separates the data points into different classes. It is recognized for its effectiveness in high-dimensional spaces, which is characteristic of text classification problems. Additionally, SVM can also handle imbalanced datasets, which is common in sentiment analysis tasks where there may be more positive or negative reviews than neutral reviews.

BiLSTM is a type of recurrent neural network (RNN) that can capture long-term dependencies in sequential data, making it particularly suitable for sentiment analysis in text. BiLSTM was chosen for its capacity to process text data in both forward and backward directions, allowing it to understand context from both before and after a word. This feature is critical in sentiment analysis, as the sentiment of a sentence often depends heavily on its overall context.

DistilBERT is a smaller, faster, cheaper and lighter version of BERT, the state-of-the-art model for a variety of natural language processing tasks [9]. It was chosen for its ability to understand the nuances of human language, including the context and semantics of words. It has been pre-trained on a large corpus of text and can therefore generate rich, contextualised embeddings for text, which can significantly improve the performance of sentiment analysis.

B. Techniques

When working with simpler models, cross-validation was used to estimate the performance of the models and to prevent overfitting. Specifically, 10-fold cross-validation was implemented, where the dataset was divided into 10 subsets and the model was trained 10 times, each time using a different subset as the validation set. This approach ensures that all data points are used for both training and validation, providing a robust estimate of the model's performance.

Early stopping is a regularization method used to avoid overfitting when training a learner with an iterative method, such as gradient descent. Training was stopped as soon as the validation error increased, or after a fixed number of iterations, to ensure that the model generalizes well to unseen data.

Hyperparameter tuning was performed to optimize the performance of the simpler models. Grid search was used to explore the hyperparameter space, and the best performing hyperparameters were chosen based on the cross-validation results.

Ensemble learning was used to improve the performance and robustness of the models. By combining the predictions of several models, ensemble methods can often achieve better performance than any single model. This is because different models may excel in different parts of the input space, and ensemble methods can leverage this diversity to make more accurate predictions.

Local Interpretable Model-agnostic Explanations (LIME) were used to interpret the predictions of the Logistic Regression model [10]. By approximating the decision boundary

of the model locally around a prediction, LIME can provide insight into which features were most influential in making the prediction. This is valuable for understanding how the model works and for identifying any potential biases or errors.

Pre-trained word embeddings were used to represent the text data. By mapping words to high-dimensional vectors that capture their semantic meaning, pre-trained word embeddings can significantly improve the performance of text classification tasks. In this project, GloVe embeddings were used due to their ability to capture both syntactic and semantic word relationships [11].

Non-negative Matrix Factorization (NMF) was used for dimensionality reduction of the TF-IDF vectors in one of the conducted experiments. By decomposing the high-dimensional vectors into a smaller number of non-negative factors, NMF can identify latent topics in the text data, which can be useful for sentiment analysis.

The Bag-of-Words and TF-IDF vectorization methods were used to convert the text data into numerical feature vectors that can be used by the machine learning models. Bag-of-Words represents each document as an unordered collection of words, disregarding grammar and word order but keeping multiplicity. It was chosen due to its simplicity and effectiveness in many text classification tasks. TF-IDF (Term Frequency-Inverse Document Frequency), on the other hand, weights each word by its importance in the document and the entire corpus, providing a more nuanced representation of the text data.

Synthetic Minority Over-sampling Technique (SMOTE) was used to balance the dataset. It works by creating synthetic samples from the minor class instead of creating copies. This technique was selected to prevent the models from being biased towards the majority class, which can often be a problem in machine learning tasks.

The Friedman test was used to compare the performance of the models. The Friedman test is a non-parametric statistical test that is used to compare more than two paired samples and can be used to conclude if there is a significant difference between the performance of the models. If the p-value is below the chosen significance level, it is possible to reject the null hypothesis and state that there is a significant difference between the accuracies of the models.

Feature engineering, extraction, and selection were also used in one of the experiments of this project. Feature engineering involved creating new features from the existing data that could help improve the performance of the models. Feature extraction, on the other hand, involved transforming the text data into a format that the models can understand, such as TF-IDF vectors or word embeddings. Finally, feature selection involved identifying the most informative features to include in the model, which can reduce overfitting and improve the interpretability of the model.

VI. Experimental setting

A. Experimental Design

The experimental design of this project was meticulously constructed to ensure a comprehensive exploration of various models and techniques suitable for the sentiment analysis task. This exploration was carried out in a step-wise manner, beginning with relatively simpler models and gradually progressing towards more sophisticated, state-of-the-art models.

The first set of models that were utilized included Naive Bayes, Logistic Regression, and Support Vector Machine (SVM). These models, although simpler in comparison to other advanced models, are widely recognized in the field of Natural Language Processing (NLP) for their effectiveness in text classification tasks [1]. The choice to begin with these models was driven by the need to establish a baseline performance level, against which the performance of more advanced models could be compared.

Three distinct experiments were conducted using these models, each with a unique approach:

Experiment 1: This initial experiment was designed to be straightforward and fundamental, employing TF-IDF vectorization to convert text data into numerical form, enabling the models to process the data. A 10-fold cross-validation was incorporated to ensure a robust evaluation of the model's performance. Furthermore, hyperparameter tuning was applied using grid search to optimize the model's parameters and improve its predictive power.

Experiment 2: Building upon the first experiment, this experiment introduced dimensionality reduction through Non-negative Matrix Factorization (NMF). This technique was employed with the intention of reducing computational complexity and improving model performance. Additionally, to address potential issues of imbalanced data, Synthetic Minority Over-sampling Technique (SMOTE) was utilized.

Experiment 3: The third experiment further extended the first experiment by adding feature engineering. Additional features such as NormalizedReviewLength (length of the review text normalized by the maximum length), ExclamationMarks (count of exclamation marks in the review text), QuestionMarks (count of question marks in the review text), and PercentageCapitalized (percentage of capitalized letters in the review text) were created to provide more informative data to the models. Moreover, Bag-of-Words was used for feature extraction alongside the TF-IDF values of the cleaned reviews. MinMaxScaler and MaxAbsScaler were applied for feature scaling, and finally, feature selection was performed using mutual information to select the most relevant features.

Building upon the best performing experiment, further enhancements were introduced. Ensemble Learning methods - Voting Classifier, Bagging Classifier, and Boosting Classifier - were employed to combine the predictions of the three models, aiming to leverage the strengths of each model to improve overall performance. Local interpretation of models using LIME was also incorporated to gain insights into which features (words) in the text most influenced the model's prediction. This not only improved the model's interpretability but also provided valuable insights for further improvements. Lastly, a Friedman test was conducted to statistically validate the difference in the performance of the models.

After this thorough exploration of simpler models, the focus was shifted towards state-of-the-art models. Two variants of the bi-directional LSTM (biLSTM) model were trained: one with pre-trained GloVe embeddings and another without

(baseline model). Pre-trained GloVe embeddings were utilized to leverage the semantic relationships that these embeddings encapsulate which have been learned from large corpora of text. The baseline model, on the other hand, allowed for a comparison of performance when such embeddings are not used.

Finally, the DistilBERT model was employed, which is a transformer-based model known for its exceptional performance in NLP tasks, including sentiment analysis. This model was chosen to evaluate whether the use of more advanced, transformer-based models could further improve the sentiment analysis performance for this specific task.

B. Data Preprocessing

Data preprocessing is an essential step in any machine learning project. It involves cleaning and transforming raw data into a format that can be understood and used by machine learning algorithms. This process helps to remove noise and inconsistencies, improve the accuracy of the models, and ultimately, enhance the performance of the machine learning system.

In this project, the initial dataset, loaded into Google Colab from a CSV file stored on Google Drive, underwent several cleaning steps. The first step involved handling missing values. Given the abundance of data available in this case, rows containing missing values were dropped. This decision was made to maintain the integrity of the dataset without resorting to estimation techniques that could introduce bias.

Next, the review text was cleaned of any superfluous elements that could impede the understanding of the sentiment conveyed. The cleaning process involved several steps, implemented using the re and nltk libraries. HTML tags and digits, typically not contributing to the sentiment, were removed from the reviews. The review text was converted to lowercase to ensure uniformity and eliminate any potential discrepancies arising from case sensitivity. Lastly, stopwords, i.e., frequently used words such as 'the', 'is', and 'and', were removed. These words, while essential for human language, often do not carry significant information useful for machine learning models and can add unnecessary complexity.

Having cleaned the textual data, the next step was to generate target labels for the sentiment analysis. The 'Rating' column of the dataset, which represents the rating given by a customer to a product, was used for this purpose. The ratings were converted into sentiment labels ('negative', 'neutral', 'positive') based on their values. Ratings below 3 were classified as 'negative', equal to 3 were considered 'neutral', and greater than 3 were deemed 'positive'. These labels were stored in a new column, 'Sentiment', which would serve as the target variable for the machine learning models.

The experiments involving biLSTM and DistilBERT also involved tokenization and padding, which are crucial for preparing text data for machine learning models, especially when using deep learning algorithms.

Tokenization is the process of breaking down the text into individual words or 'tokens'. In the context of this project, the review texts were tokenized into separate words. This process enables the models to understand and learn from the text data by treating each word as a discrete entity.

However, this leads to varying lengths of token sequences, as reviews can be of different lengths. To feed these sequences into the machine learning models, they need to be of the same size. This is where padding comes in. Padding refers to the process of standardizing the length of input sequences by appending zeros to the shorter sequences. This ensures that all input sequences to the model have the same shape, which is a requirement for most neural network architectures.

C. Synthetic Minority Over-sampling

Data augmentation techniques are often employed to increase the robustness and generalization capabilities of Machine Learning models. In the context of imbalanced datasets, these techniques can be particularly useful, as they help to mitigate the bias towards the majority class that can be introduced during model training.

In the case of this project, the original dataset exhibited a substantial imbalance, which is illustrated in table I and figure 1.

Class	Instances
Positive	284997
Negative	97078
Neutral	31765

TABLE I: Data Distribution

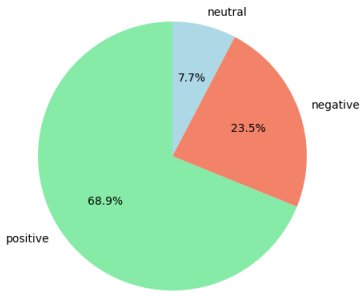

Fig. 1: Pie chart of data distribution.

To address this imbalance, the Synthetic Minority Oversampling Technique (SMOTE) was employed in the second experiment involving Naive Bayes (NB), Logistic Regression (LR), and Support Vector Machines (SVM).

SMOTE is a well-established preprocessing technique used to tackle class imbalance in datasets [12]. It operates by creating synthetic examples of the minority class, thereby balancing the class distribution. Specifically, SMOTE selects random instances of the minority class, identifies their k-nearest neighbors, and generates synthetic instances along the line segments connecting the selected instances to their neighbors.

The primary advantage of using SMOTE is its potential to improve the performance of ML models on imbalanced datasets. By balancing the class distribution, SMOTE enables the models to learn more effectively from the minority class, which can be particularly beneficial in a sentiment analysis

context where negative and neutral reviews may be less frequent but equally important.

However, it's important to note that SMOTE may not be the optimal solution for every imbalance problem. The synthetic examples it creates are based on linear interpolation between existing instances, which might not always capture the underlying distribution of the minority class.

D. Local Interpretable Model-agnostic Explanations

In the course of this project, the technique of Local Interpretable Model-agnostic Explanations (LIME) was employed to provide insight into the decision-making process of the employed machine learning models [10]. The choice for LIME, in this case, pivoted on the crucial role of explainability in Artificial Intelligence and Machine Learning. The general trend in machine learning has been to produce increasingly complex models that achieve high predictive performance at the expense of interpretability. This trade-off has led to the creation of so-called "black box" models, which make accurate predictions but do not provide clear explanations for their decisions.

However, in many applications, including sentiment analysis, the ability to understand the rationales behind the decisions of machine learning models is not merely an academic interest but rather a critical requirement. This is especially pertinent when the results of the machine learning model will be used to inform business decisions or when the model's potential misclassification could have significant consequences. Thus, the incorporation of LIME serves as a bridge to understanding the often opaque decision-making process of machine learning models.

LIME operates by approximating the decision boundary of a complex machine learning model with a simpler, interpretable model. In particular, the algorithm generates a set of perturbed instances from the original data point and trains a simpler model on these perturbed instances. This simpler model, often a linear model, is then used to elucidate the prediction of the original complex model by highlighting the most salient features that contributed to the prediction.

An important attribute of LIME is its local nature, providing explanations for individual predictions rather than global explanations for the entire model. This granularity is particularly beneficial when the aim is to diagnose and understand the behavior of machine learning models in specific instances, as was the case in this project.

In the scope of this project, LIME was utilized on a single instance with the Logistic Regression model, which performed optimally among all the simple baseline models developed. The original review text for the instance is as follows:

> "Well first off I was sent the wrong phone which put me in a tight.. Once I finally received the correct phone it seemed to work like a dream . It had pretty much all the memory to be refurbished which was great. Then I noticed that the battery was no good.. it could barely last over an hour with average use so now I'm going to have to buy a battery. Then (of course I haven't dropped it) the screen is not in its frame.. it's popping up and so I'm going to have to pay for someone to put it in correctly .. this is very hectic and annoying."

The associated customer rating for the review was 3, which, given the task formulation in this project, assigns the instance a true label of "neutral". However, intriguingly, the model categorized the review as negative. This discrepancy between the model's prediction and the true label prompted further investigation using LIME.

The analysis with LIME highlighted the significance of specific words in the review text, thereby offering insight into why the model classified the review as negative. A visual representation of the LIME output, which color-codes the words based on their contribution to the final prediction, is provided in Figure 2.

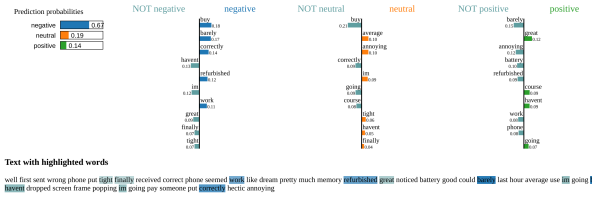

Fig. 2: LIME output.

Through the lens of LIME, it is evident that the negative sentiment expressed in certain parts of the review, such as the mention of the wrong phone being sent initially, the poor battery life, and the screen issue, significantly influenced the model's decision to classify the review as negative.

The insights garnered from the application of LIME on this instance serve a dual purpose. On one hand, they provide a rationale for the model's classification of the review as negative, enhancing the understanding of the model's decision-making process. On the other hand, they stimulate a reevaluation of the task formulation itself. Specifically, the instance under consideration raises the question of whether a ternary classification (positive/negative/neutral) is the optimal way to model sentiment analysis for this dataset. Given the complexities and nuances inherent in human sentiment, it might be beneficial to consider a binary classification (positive/negative) instead, particularly as it appears that a neutral rating can encompass a range of sentiments.

This reflection underscores the value of LIME, not only as a tool for model interpretation but also as a catalyst for critical thinking about model design and problem formulation. Thus, even though LIME was applied to a single instance in this project, its contribution to the project was profound, highlighting the potential for its broader application in future work.

VII. Experimental Results

This section showcases the outcomes of the various experiments performed to asses the efficacy of different models in conducting sentiment analysis on product reviews. Table II lists the performance metrics derived for each experiment, while Figure 3 visually depicts the comparative performance of the different models.

Model	Accuracy	Precision	Recall	F1-score
Naive Bayes	0.85	0.70	0.60	0.61
Logistic Regression	0.87	0.76	0.67	0.68
SVM	0.87	0.77	0.65	0.66
biLSTM with GloVe	0.92	0.85	0.81	0.82
biLSTM without GloVe	0.92	0.85	0.82	0.83
DistilBERT	0.93	0.86	0.84	0.85

TABLE II: Comparison of Experimental Results

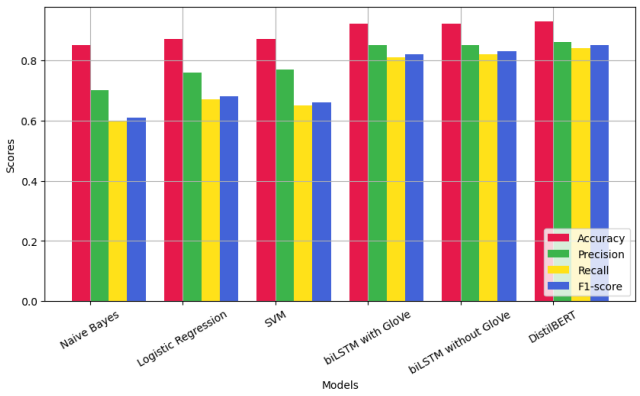

Fig. 3: Grouped bar chart of experimental results.

The models Naive Bayes, Logistic Regression, and Support Vector Machines (SVM) were trained using 10-fold cross-validation. This approach enabled the execution of a statistical test to determine if the performance of these models differed in a statistically significant way. For this purpose, the Friedman test was employed and the obtained results are presented in Table III. A box plot illustrating the comparison of these models is displayed in Figure 4.

Statistic	**Value**
F statistic	20.0
p-value	4.539992976248486e-05

TABLE III: Friedman Test Results.

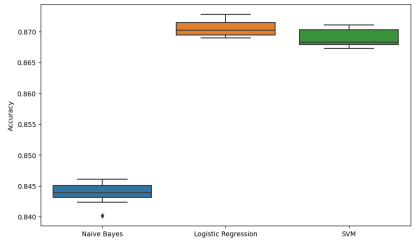

Fig. 4: Box plot of simple model accuracy.

In the context of ensemble learning, Table IV enumerates the accuracy values obtained for the three implemented ensemble methods. Figure 5 provides a visual comparison of their performance.

Ensemble Method	**Accuracy**
Voting Classifier	0.87
Bagging Classifier	0.87
Boosting Classifier	0.80

TABLE IV: Ensemble Learning Results.

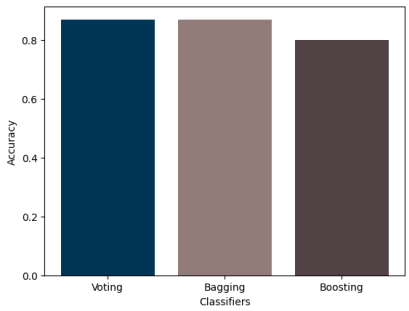

Fig. 5: Ensemble method accuracy comparison.

VIII. Result Analysis

A. Performance Analysis

The Naive Bayes model achieved an accuracy of 0.85, with precision, recall, and F1-score results of 0.70, 0.60, and 0.61 respectively. Its performance, while lagging behind the other models, was still considerably robust and could be seen as a benchmark for the performance of other more sophisticated models.

In comparison, the Logistic Regression and SVM models displayed higher accuracies at 0.87. Their results in terms of precision, recall, and F1-scores were also closely aligned, suggesting similar predictive capabilities. However, the Logistic Regression model exhibited a slightly higher recall, indicating a potentially superior ability in identifying true positive cases, a crucial aspect in sentiment analysis tasks.

The biLSTM models, with and without the GloVe embeddings, outperformed the previous models, achieving an accuracy of 0.92, with precision, recall, and F1-score results in the range of 0.81 to 0.85. Intriguingly, the performance difference between the two biLSTM models was minimal, implying that pre-trained embeddings might not always provide a substantial performance enhancement in sentiment analysis tasks.

The highest performance was observed in the DistilBERT model, a transformer-based model known for its powerful context-aware text representation capabilities. This model exhibited an accuracy of 0.93, precision of 0.86, recall of 0.84, and F1-score of 0.85. These results suggest that transformer-based models are possibly more adept at capturing complex patterns in the text data, making them potent tools for sentiment analysis tasks.

As for ensemble methods, the Voting and Bagging Classifiers both achieved an accuracy of 0.87, surpassing the Boosting Classifier which had an accuracy of 0.80. Despite their lower accuracy compared to some individual models, these ensemble methods still contributed valuable insights into the potential benefits of using multiple models in concert.

B. Discussion

The analysis of the experimental results revealed several interesting patterns and insights. As anticipated, the state-of-the-art models, the biLSTMs and DistilBERT, demonstrated the highest performance. However, the performance of the simpler models was also noteworthy, exceeding initial expectations.

The performance of the biLSTM models, specifically the nearly equivalent results with and without the GloVe embedding, was unexpected. One possible explanation could be that the Amazon product reviews dataset used in this project contained specific jargon, slang, or abbreviations that were not effectively captured by the GloVe embeddings, which are trained on a different corpus. To address this, it might be beneficial to use word embeddings trained on a similar product reviews corpus, ensuring the embeddings are more suited to the specific language used in this context.

Interestingly, the performance of the simple models was more robust even the use of advanced techniques such as feature engineering and selection, dimensionality reduction, or synthetic minority over-sampling. This could be due to the inherent characteristics of the dataset. In certain scenarios, adding complexity through feature manipulation might introduce noise or overfitting, reducing the model's ability to generalize.

The results of the Friedman test, conducted to validate the observed differences in performance among the Naive Bayes, Logistic Regression, and SVM models, indicated a statistically significant difference in the performance of the models.

Another intriguing finding was the comparable performance of the biLSTM and DistilBERT models. While it was expected that the transformer-based DistilBERT model would significantly outperform the biLSTM due to its more advanced architecture, the experimental results did not reflect this. This could potentially be attributed to the nature of the sentiment analysis task, which might not require the complex pattern recognition capabilities of the DistilBERT model. Alternatively, it could be a result of the DistilBERT model not being fully exploited, possibly due to the choice of hyperparameters or the training procedure.

Finally, the performance of the ensemble methods, while not exceeding that of the best individual models, still provided valuable insights. The differences in performance among the Voting Classifier, Bagging Classifier, and Boosting Classifier suggest that certain ensemble techniques may be more effective than others for this specific task. Further research could be done to explore the combination of different models in the ensemble, or the application of more advanced ensemble methods.

IX. Future work

The outcomes and insights gathered from this project offer numerous ideas for future work. These ideas primarily center around the following themes: refining and extending the current models, experimenting with different types of pre-processing and feature engineering, and exploring other advanced models and techniques.

The first theme pertains to the refinement and extension of the models used. While the performance of the models was robust, there is still room for improvement. For instance, the potential of the DistilBERT model might not have been fully realized, as suggested by its comparable performance with the simpler biLSTM models. More extensive hyperparameter tuning or different training procedures could be explored to fully leverage the capabilities of this transformer-based model. Additionally, the biLSTM models could be further refined by experimenting with different architectures, such as adding more layers or neurons, or using different types of recurrent layers.

The second theme revolves around the preprocessing and feature engineering techniques employed. The performance of the simpler models was robust even without advanced feature extraction, engineering, selection, or dimensionality reduction techniques. However, this does not rule out the possibility that other preprocessing or feature engineering methods could lead to improvements. For instance, different methods of text cleaning or the use of more sophisticated feature extraction techniques such as Word2Vec or FastText could be investigated. Similarly, different approaches to dimensionality reduction or feature selection could be employed to further refine the feature set used by the models.

The third theme concerns the exploration of other advanced models and techniques. The performance of the ensemble methods, while not exceeding that of the best individual models, suggested that there could be benefits in using multiple models in concert. Building on this, future work could investigate the use of more advanced ensemble methods. Additionally, other state-of-the-art models, such as other transformer-based models, could be employed to further enhance the performance of the sentiment analysis task. Lastly, more advanced interpretation techniques could be used to gain deeper insights into the model's predictions, thereby improving the interpretability of the models.

X. Conclusion

This project conducted a comprehensive exploration of various models and techniques for the sentiment analysis task on Amazon product reviews. A meticulous experimental design was established, initiating with simpler models to set a baseline performance level, and then progressively transitioning towards more sophisticated, state-of-the-art models. The experiments provided valuable insights into the performance of different models and techniques and identified several areas for potential future work.

The project's results indicate that both simple and advanced models can offer robust performance in sentiment analysis tasks. Notably, the biLSTM and DistilBERT models demonstrated the highest performance, although simpler models like Naive Bayes, Logistic Regression, and SVM also performed robustly.

The findings also revealed intriguing patterns that could inform future research. For example, the nearly equivalent performance of the biLSTM models with and without the GloVe embeddings suggests that using pre-trained word embeddings might not always offer performance enhancements. Similarly, the comparable performance of the biLSTM and DistilBERT models suggests that advanced architectures might not always outperform simpler ones in sentiment analysis tasks.

Furthermore, the ensemble methods provided valuable insights into the potential benefits of using multiple models in concert. Although the ensemble methods did not exceed

the performance of the best individual models, they still demonstrated the potential for improving performance through model combination.

In conclusion, this project has provided a thorough and insightful exploration of sentiment analysis in Amazon product reviews. The findings have contributed to the understanding of the strengths and weaknesses of various models and techniques and have provided a foundation for future work in this area.

REFERENCES

[1] M. Ezhilarasan, V. Govindasamy, A. Gopu, and K. Vadivelan, "Sentiment analysis on product review: A survey," 03 2019, pp. 180–192.

[2] R. Ni and H. Cao, "Sentiment analysis based on glove and lstm-gru," 07 2020, pp. 7492–7497.

[3] H. Elfaik and E. H. Nfaoui, "Deep bidirectional lstm network learning-based sentiment analysis for arabic text," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 395–412, 2021. [Online]. Available: https://doi.org/10.1515/jisys-2020-0021

[4] K. Zhou and F. Long, "Sentiment analysis of text based on cnn and bi-directional lstm model," in *2018 24th International Conference on Automation and Computing (ICAC)*, 2018, pp. 1–5.

[5] M. Arbane, R. Benlamri, Y. Brik, and A. D. Alahmar, "Social media-based covid-19 sentiment classification model using bi-lstm," *Expert Systems with Applications*, vol. 212, p. 118710, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422017353

[6] S. N. Almuayqil, M. Humayun, N. Zaman, M. Almurafeh, and N. A. Khan, "Enhancing sentiment analysis via random majority undersampling with reduced time complexity for classifying tweet reviews," *Electronics*, vol. 11, p. 3624, 2022.

[7] M. Kuhn and K. Johnson, *Feature Engineering and Selection: A Practical Approach for Predictive Models*, 1st ed. Chapman and Hall/CRC, 2019. [Online]. Available: https://doi.org/10.1201/9781315108230

[8] PromptCloud, "Amazon reviews - unlocked mobile phones," https://www.kaggle.com/datasets/PromptCloudHQ/amazon-reviews-unlocked-mobile-phones, 2016, accessed: July 2023.

[9] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 10 2019.

[10] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 2016, pp. 1135–1144.

[11] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in *EMNLP*, vol. 14, 2014, pp. 1532–1543.

[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.