Amazon Product Review Classification using Sentiment Analysis

CSCI5502 - Final Project Report

Dhanavikram Sekar, Hariharan Kumar, Naveen Vinayaga Murthy

December 2023

Abstract

In today's rapidly evolving e-commerce sector, the abundance of user-generated content, particularly in the form of product reviews, requires robust natural language processing approaches to extract meaningful insights. This project focuses on exploration of sentiment analysis techniques to identify the nuanced sentiments within the product reviews collected from Amazon website. Using this information, The sellers could improve their product experience in response to customer preferences and formulate targeted product development strategies. This report will go in-depth about the methodologies followed, including data collecting strategies, preprocessing methods, and the reasoning behind the machine learning algorithm choices.

1 Introduction

1.1 Motivation

In the current digital marketplace, the reviews posted online by consumers play a major role in influencing the sales of a product. One of the primary ways users decide on buying a product is by looking at the positive and negative aspects of a product. But does a consumer have to go through all the reviews to identify such aspects? This endeavor holds immense value for such consumers as it specifically allows them to read a review of a particular sentiment of their choice and thus identify the positive and negative aspects of the product from those reviews. This can also be used by sellers who aim to improve their product quality and user experience based on user feedback.

Column name	Description	Selection
reviewerID	ID of the reviewer	Not Selected
asin	ID of the product	Not Selected
reviewerName	name of the reviewer	Not Selected
vote	helpful votes of the review	Not Selected
style	a dictionary of the product metadata	Not Selected
reviewText	text of the review	Selected
overall	rating of the product	Selected
summary	summary of the review	Selected
unixReviewTime	time of the review (unix time)	Not Selected
reviewTime	time of the review (raw)	Not Selected
image	images that users post after they have received the product	Not Selected

Table 1: List of columns and their description

1.2 Objective

This project focuses on extracting valuable insights from the abundance of such textual consumer feedback available for a certain category of products in Amazon's e-commerce website. The technical objective of this project is to develop a robust machine learning model that can efficiently classify product reviews into positive, negative and neutral sentiments using Data Mining, Sentiment Analysis and Natural Language Processing techniques. This report's goal is to provide a comprehensive overview of sentiment analysis techniques used to classify product reviews and a methodical presentation of findings and a detailed discussion of their implications.

2 Methods

2.1 Data Collection and Description

The dataset utilized in this sentiment analysis project was published in Ni et al. (2019). This dataset is an archive of customer reviews for various product categories scraped from the Amazon website. For this particular project, the Luxury product category is chosen out of all the available categories. The primary reason for choosing one particular category is the constraint on the available computing resources. The dataset comprises textual reviews submitted by each user as well as the metadata associated with them, making it appropriate for training and evaluating sentiment analysis based classification models.

Table 1 provides an overview of the features that are present in the collected dataset. The first column lists the variable names. The second column offers detailed descriptions of each variable. The third column indicates whether a particular variable has been selected for inclusion in the analysis and modeling phase.

2.2 Data Pre-processing

In the pre-processing phase, two libraries Spacy and NLTK are used and compared, particularly in tasks such as stopword removal and lemmatization. Figure 1 gives an overall view of the data preprocessing pipeline.

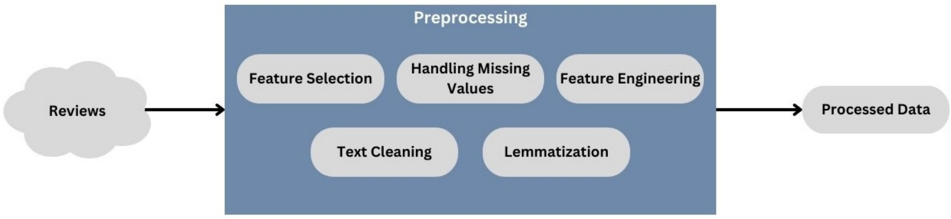

Figure 1: Preprocessing Pipeline

2.2.1 Feature Selection

The variables chosen for in-depth analysis and modeling include 'reviewText', 'summary', and 'overall'. The 'reviewText' column was specifically selected as it contains all of the review contents. The 'summary' column, representing the review title, was also included as it often contains key words indicative of the review's sentiment. The 'overall' column was also selected as it is crucial for data labeling purposes, serving as an indicator of sentiment. This enables defining the problem as a supervised classification problem.

2.2.2 Handling of Missing values and Duplicates

Thorough examination of the dataset uncovered a small number of missing values, comprising only 0.08% of the data. Another analysis revealed that 28% of the data are duplicated observations. To improve the quality of data, the rows containing null values and duplicates were removed.

2.2.3 Feature Engineering

The text columns 'reviewText' and 'summary' have been merged together to form a single textual column, as it allows representing information from multiple variables in a single variable and easier vectorization.

The 'overall' column has been used to create labels for the reviews. This particular column represents the star ratings given by a user for a particular product and so it might represent the sentiment of the user towards a particular product. On this basis, observations with star ratings greater than 3 were considered positive and those with star ratings under 3 have been considered negative. The reviews with exactly 3 star ratings are considered as neutral reviews. To align with the numerical requirements of machine learning models, positive reviews are labeled as 0, neutral reviews as 1 and negative reviews as 2.

2.2.4 Text Cleaning

To eliminate special characters and punctuation from each sentence, regex methods were applied. Python's inbuilt functions such as *lower* have been used to convert all the text into lower case.

Words that do not carry any contextual meanings in the modeling stage are called stopwords. Stopwords might be articles, pronouns, words such as this, that, etc., Both Spacy and NLTK have been used to remove stopwords. However, one important finding is that Spacy often removed words that represent numbers like 'one' and 'five'. In this particular task these words might indicate the sentiment of the review. This is further discussed in the final section.

2.2.5 Lemmatization

Lemmatization involves reducing a word to its base or root form. This ensures consistency in the representation of words and contributes to a more coherent understanding of sentiments by a machine learning model. For lemmatization, Spacy's inbuilt lemmatizer and NLTK's WordNet Lemmatizer has been used and compared. To facilitate NLTK's lemmatization process, each of the words in the review has been tagged specifically with appropriate parts of speech using NLTK's inbuilt parts of speech tagging function.

2.2.6 Spacy vs NLTK

This project also compares the performance of the preprocessing libraries used. Spacy being a library based on C-based Python called Cython is supposed to perform better compared to NLTK library which is based on Python's inbuilt string library. On comparing the execution time taken for preprocessing the textual data based on the aforementioned structure in Figure 1, Spacy was found to be 30% faster compared to NLTK. However besides performance, the quality of data that is produced along with the performance of models on the text preprocessed using these libraries has also been assessed.

2.3 Data Analysis

An in-depth detailed analysis has been performed after the preprocessing stage. The analysis results and the reasoning behind choosing a certain type of visualization has been explained in the following sections.

2.3.1 Size of Classes

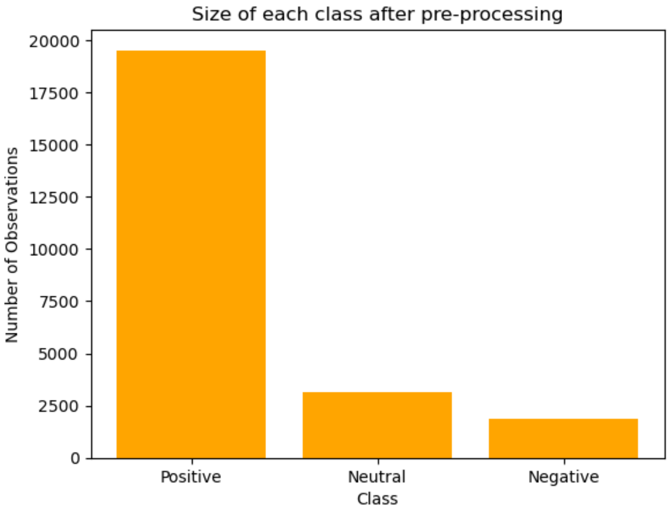

Figure 2: Class Imbalance Problem

A bar plot has been plotted to visualize the size of each class after preprocessing. Figure 2 shows a clear class imbalance problem in the dataset. The number of positive reviews are high, followed by neutral and then negative reviews. Techniques used to solve this class imbalance problem are discussed in later sections.

2.3.2 Histogram of word count

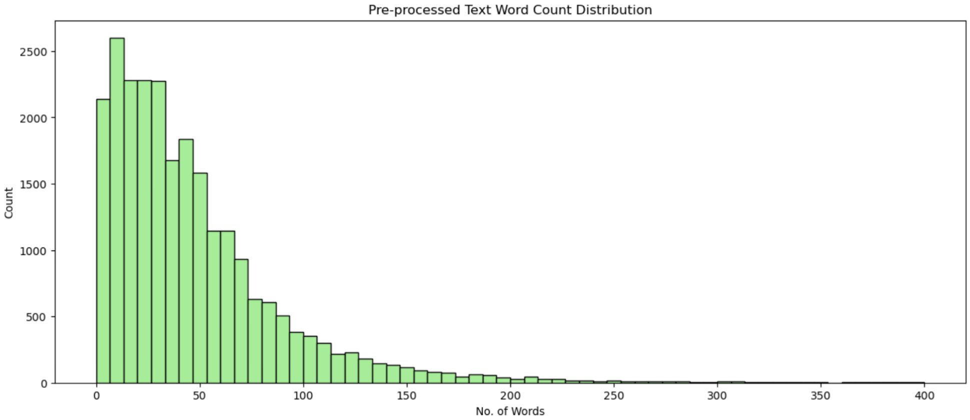

Figure 3: Word count Distribution

A histogram has been plotted to find the distribution of the count of words in the reviews. Figure 3 shows a right-skewed distribution, with most of the reviews having a word count between 0 and 200 words, and a smaller tail extending out to reviews with over 400 words.

1. Most reviews are relatively short. This suggests that customers are more likely to write brief summaries of their experiences rather than in-depth essays.
2. There is a small but significant number of longer reviews. These could be from customers who are passionate about the product and have had a particularly positive or negative experience, or are writing for a specific audience, such as a product review blog.

2.3.3 Word Cloud

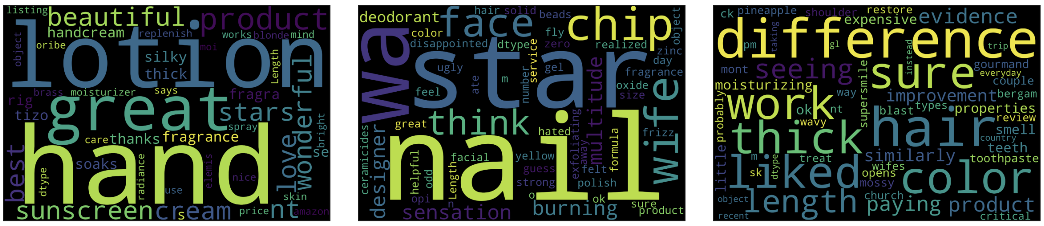

Figure 4: Word Clouds for Positive, Neutral and Negative Reviews

Figure 4 illustrates the most frequently occuring words in each of the classes. The left most word cloud represents the frequent words in positive reviews. It is discernible that words like 'hand', 'lotion', 'sunscreen' do not carry any positive meaning. Similarly for neutral reviews in the middle and for negative reviews in the right, there are certain words like 'nail', 'chip', 'thick', 'hair', etc., which do not convey any sentiment related information. The weightage to these words can be reduced in the sentiment analysis process by choosing the right vectorization metric.

2.4 Vectorization

Vectorization involves converting text into a numerical form so that it can be parsed and learnt by a machine learning model. For this specific project, two methods were chosen.

2.4.1 Bag of Words

Bag of Words model represents the text in a numerical form by considering the frequency of a word in a single review. It creates a vocabulary of all the unique words present in the data and assigns each word a unique index. Each review is then represented by a vector where each element in the vector represents that count of that particular index in the vocabulary in an unordered manner.

2.4.2 TF-IDF

TFIDF (Term Frequency - Inverse Document Frequency) is an extension of Bag of Words which improves the representation by taking into account every word's frequency in a single observation and in the entire data. TF(Term Frequency) measures the frequency of a word in a single review and IDF(Inverse Document Frequency) measures its frequency in the entire data. Ramos et al. (2003) discusses the significance of assigning weights to words based on their frequency. Thus in a TF-IDF technique, a word is assigned higher weightage if it appears often in a single review and appears rarely in the entire vocabulary. So less frequent characters get higher weightage.

2.4.3 Choosing n-grams

The disadvantage of both these methods is that they do not consider the order of words, which might contain contextual information. This problem can be solved by creating a vocabulary of n-combination of words. Tripathy et al. (2016) in their paper, have compared the performance of different values of n for a text classification problem. Similar to their approach, this project incorporates a heuristic approach for choosing the value of the hyperparameter n for n-grams for both Bag of Words and TF-IDF methods.

On running a model with different combinations of values for n-grams in vectorization techniques, a combination of unigram(n=1) and trigram(n=3) has been chosen, as it yielded the best results.

2.5 Solving the class imbalance problem

On analysis of the data, it was evident that the number of positive reviews was very high compared to the number of neutral and negative reviews. Figure 2 illustrates the distribution of the number of observations for each of the classes.

To avoid the model over fitting on one specific class and under performing on the other two classes, it is essential to solve this class imbalance problem. To solve this, two techniques can be used. One technique is undersampling of the classes with a high number of observations. The other technique involves oversampling or increasing the size of the classes with low number of observations artificially to match the size of the majority class. As demonstrated in Chawla et al. (2002), a combination of undersampling of the majority class and oversampling using SMOTE(Synthetic Minority Oversampling Technique) can be used to improve the performance of the classification models. MOHASSEB et al. (2018) in their paper, have also illustrated the performance of SMOTE in a text classification problem. Hence, SMOTE based oversampling of minority classes in combination with undersampling of majority class has been utilized in this venture. Instead of generating all the synthetic samples upfront, SMOTE is integrated within a cross-validation pipeline, allowing it to dynamically create new samples during each fold of the cross-validation process.

2.6 Modelling

Figure 5 gives an overview of the model pipeline used to classify reviews based on sentiments.

2.6.1 Models Chosen

Rain (2013) in their paper have demonstrated the use case of probabilistic machine learning in the sentiment analysis domain. **Naive Bayes** is a classification algorithm based on Bayes theorem. It is based on the assumption that the features that are given as input are conditionally independent. Prior probabilities and likelihood probabilities are calculated and the results are predicted based on the highest posterior probability. This model is specifically chosen as it calculates the posterior probability for a particular review belonging to a particular sentiment with the calculated prior and

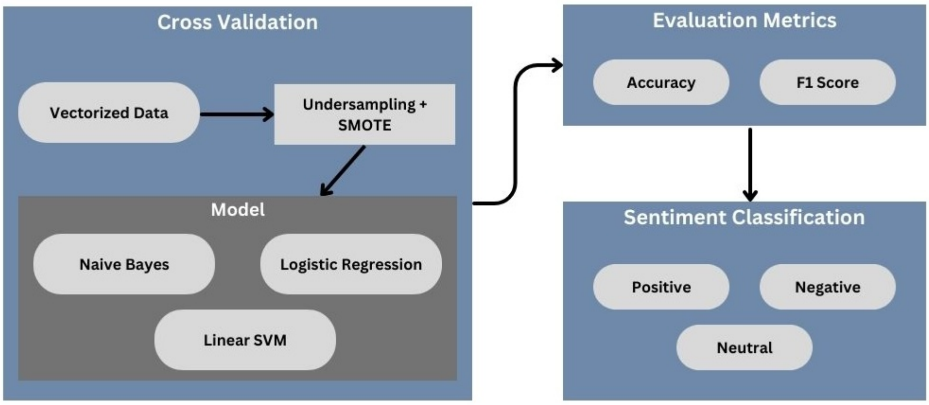

Figure 5: Model Pipeline

likelihood probabilities calculated based on the training data. Multinomial Naive Bayes theorem has been used as this problem involves multi-class classification.

Logistic Regression is one of the other classification algorithms chosen because of its versatility in the classification problems. Multinomial or softmax regression based Logistic Regression is chosen in which the model is trained with multiple output classes and uses softmax function to calculate class probability. The model is based on maximum likelihood and the class with highest probability is chosen as the final classified label. To increase the computational efficiency, Logistic Regression coupled with Stochastic Gradient Descent Optimization algorithm has been used.

Linear Support Vector Machines finds the optimal linearly separable hyperplane to classify the data points. If the number of features is more, Linear SVM is very effective. In this project, since the dimensions of the vectorized data is very high, Linear SVM was chosen. The superior performance of this model is elaborated in Rathor et al. (2018).The easy interpretability of this model is another reason to choose this model. The predictions are made based on which side of the hyperplane the data point falls.

The main reason for choosing these three models lie in their simplicity and efficiency, making them well versed for low-powered computational resources like laptops and personal computers.

2.6.2 Cross Validation

Cross validation is a technique which is used for enhancing the model assessment and generalization of predictive models. In this project, to achieve generalized results for different variations of data, cross validation has been carried out on a Stratified K-Fold data n-times. It repeatedly partitions the data into multiple subsets and one of the subset is used as testing data and the other subsets are used as training data. By running the cross validation pipeline on Stratified K-Fold data n times, its performance is fairly evaluated across different subsets of the same data. In this project, data has been split into 5 subsets and the entire cross validation process has been repeated 3 times for optimal and non-biased results.

3 Results

3.1 Metrics Chosen

For this specific problem, two metrics are chosen. **Accuracy** is one of the metrics chosen, as it represents the number of samples correctly classified among all the classifications. However, since there is a class imbalance problem, there is a high possibility that the model performs well on the class with high number of samples and performs poorly on the model with low sample size. Because of this, the accuracy might be really high even if the model performs poorly on the classes with a low number of samples.

To tackle this, Precision and Recall can be used, as they give a holistic view of how the model performs on a particular class. For this project **F1-Score** is the other metric chosen, as it is the harmonic mean of precision and recall values. Since this is a multi-class classification problem, weighted F1-score is calculated, as it gives the mean of the F1-score of all the classes.

3.2 Achieved Results

The scores of all the chosen models in combination with different vectorizers for Spacy and NLTK versions of preprocessed data is mentioned in Table 2. The achieved results are compared to the results of Tan et al. (2018) and the results indicate the better performance of the same models with different vectorization techniques . It is discernible that Logistic Regression and Linear Support Vector Machines in combination with TF-IDF vectorization yields the best results for both Spacy and NLTK versions of the dataset both in terms of Accuracy and F1-score.

Figure 6 displays the confusion matrices for the Logistic Regression and Linear SVM models for NLTK versions. It can be seen that although the Linear SVM performs well on positive class, it performs subpar compared to the Logistic Regression model. Similarly for the Spacy versions of the preprocessed data, Linear SVM performs well for the positive label however struggles to classify the negative and neutral labels as shown in Figure 7. This is clearly because of the class imbalance problem discussed in the pre-

Model	**Vectorizer**	**Preprocesser**	**Accuracy**	**F1-Score**
Naive Bayes	Bag of Words	Spacy	0.69	0.60
Logistic Regression	Bag of Words	Spacy	0.70	0.70
Linear SVM	Bag of Words	Spacy	0.68	0.69
Naive Bayes	TF-IDF	Spacy	0.65	0.68
Logistic Regression	TF-IDF	Spacy	0.72	0.73
Linear SVM	TF-IDF	Spacy	0.76	0.75
Naive Bayes	Bag of Words	NLTK	0.70	0.61
Logistic Regression	Bag of Words	NLTK	0.71	0.71
Linear SVM	Bag of Words	NLTK	0.70	0.70
Naive Bayes	TF-IDF	NLTK	0.65	0.67
Logistic Regression	TF-IDF	NLTK	0.75	0.75
Linear SVM	TF-IDF	NLTK	0.78	0.78

Table 2: Scores of Models in combination with different Vectorizers and Pre-processors

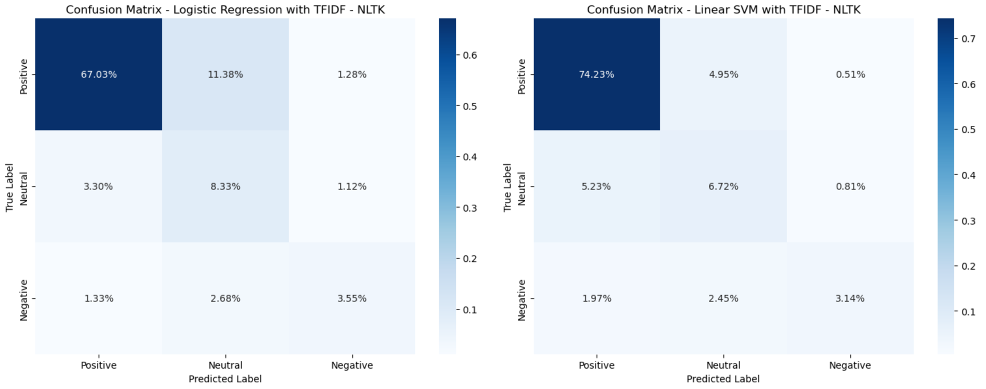

Figure 6: Confusion Matrix - NLTK

vious section.

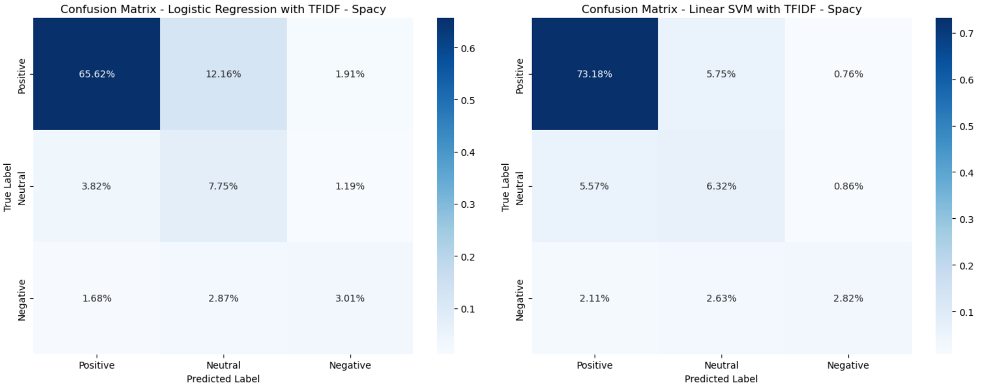

Figure 7: Confusion Matrix - Spacy

4 Conclusion

4.1 Findings

Based on the results, it was evident that Logistic Regression and Linear Support Vector Machines in combination with TF-IDF vectorization performs well in the task of classifying textual reviews with a F1-score of 0.75 and 0.78. However, even after applying a combination of under-sampling and over-sampling techniques, the class imbalance problem still poses a major hurdle in the performance of machine learning models. From Figure 6 and Figure 7, it is evident that both top performing models perform poorly on the classes with low number of samples.

Another interesting finding was that all models perform better on NLTK's version of pre-processed data compared to Spacy. The potential reasons for models performing better on NLTK version of data may be attributed to the way by which these two pre-processors eliminate stopwords. For example, the word 'Five' in a review 'Five stars. great!' plays a major role, as it indicates the number of stars and indirectly the emotion. Spacy removes these words, but NLTK does not. Similarly, there are certain stopwords which might carry meaning in the context of this particular dataset for sentiment analysis, but Spacy removes them by default .

4.2 Limitations and Future Scope

The main limiting factor for this endeavor was the sparse matrices created by the vectorization methods. The shear size of these matrices might be a contributing factor for high average execution time taken during the training of models. To overcome this, PCA and Truncated SVD techniques were implemented with the intention of reducing dimensions of the vectorized data. But no notable improvements were observed both in terms of execution time and evaluation metrics.

In future, better word embedding techniques such as Word2Vec, GloVe and BERT can be used. These techniques generate word embeddings(vectorized data) of shorter and fixed length, which can solve the sparse matrix limitations of the Bag of Words and TF-IDF methods used in this project. Katić and Milićević (2018) demonstrates the improvement in the performance of models in combination with the aforementioned word embedding techniques in comparison with the techniques used in this project. Furthermore, since these word embeddings have contextual meanings to them, SMOTE will be able to create better quality synthetic samples which in-turn can improve the performance of machine learning models.

Another limitation of this project is the choice of machine learning models. In future, with more time and knowledge, deep learning models such as LSTM(Long Short Term Memory) based Recurrent Neural Networks as proposed in Mohbey (2021) can also be employed in combination with the word embeddings to replicate state-of-the-art results that can be achieved by using pre-trained models.

References

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

Katić, T. and Milićević, N. (2018). Comparing sentiment analysis and document representation methods of amazon reviews. In *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*, pages 000283–000286. IEEE.

MOHASSEB, A., BADER-El-DEN, M., COCEA, M., and LIU, H. (2018). Improving imbalanced question classification using structured smote based approach. In *2018 International Conference on Machine Learning and Cybernetics (ICMLC)*, volume 2, pages 593–597.

Mohbey, K. K. (2021). Sentiment analysis for product rating using a deep learning approach. In *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, pages 121–126.

Ni, J., Li, J., and McAuley, J. (2019). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of* *the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197.

Rain, C. (2013). Sentiment analysis in amazon reviews using probabilistic machine learning. *Swarthmore College*, 42.

Ramos, J. et al. (2003). Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer.

Rathor, A. S., Agarwal, A., and Dimri, P. (2018). Comparative study of machine learning approaches for amazon reviews. *Procedia computer science*, 132:1552–1561.

Tan, W., Wang, X., and Xu, X. (2018). Sentiment analysis for amazon reviews. In *International Conference*, pages 1–5.

Tripathy, A., Agrawal, A., and Rath, S. K. (2016). Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, 57:117–126.