LING227 Final Project: Sentiment Analysis Classification of Amazon Product Reviews

Charlson Kim and Jin Wuk Lee

December 2024

1 Introduction

In the digital age, online product reviews have become essential for consumers navigating purchasing decisions, especially with the rise of e-commerce platforms like Amazon and Alibaba. Beyond evaluating product quality, they strongly influence decision-making, particularly for new or unfamiliar items. Thus, these reviews, filled with user sentiment and opinions, play a pivotal role in shaping buying behavior through Word-of-Mouth (WOM) communication. The growing importance of product reviews across various industries highlights their critical impact on consumer perceptions and choices. Positive online reviews, for instance, serve as "social proof," providing reassurance for customers and encouraging them to make purchases. They enhance decision-making by offering authenticity and unbiased insights, addressing specific concerns a potential buyer might have (Kang et al, 2022 [6]).

Sentiment Analysis, a subfield of NLP, is widely used to examine subjective expressions, such as opinions or attitudes toward a particular entity. While sentiment analysis is primarily applied to written text, its scope extends to other modalities, such as extracting sentiments from audio, images, or video sequences. The sentiments analyzed are typically categorized as positive, negative, or neutral, but they can also encompass a broader range of emotions or feelings as well categorical labels in the form of 1 to 5 stars (Liu, 2015 [4]). This project focuses on sentiment classification, a type of text classification that often leverages supervised learning techniques. Commonly used methods include logistic regression, transformers, support vector machines (SVM), and Recurrent Neural Networks (RNNs), all of which have demonstrated success in text classification tasks (Lu et al., 2020 [5]). Multiple NLP techniques will be employed in this exploration of Sentiment Analysis task with the predominant focus being on the bidirectional Transformer model of BERT, Logistic Regression, Random Forest Regression, and Support Vector Machines. These approaches generally rely on frequency-based text vectorizers to transform text into numerical representations for model input.

This study focuses on sentiment classification of Amazon product reviews, which are written by real customers about products sold on the platform. By comparing the predictive performance of multiple computational approaches, the study seeks to determine if the advanced capabilities of BERT translate into meaningful improvements over statistical and machine learning methods. The central research question, therefore, is: "Does fine-tuning a pre-trained BERT model significantly enhance predictive accuracy for sentiment classification of Amazon reviews compared to traditional statistical and machine learning models?"

2 Algorithms/Related Work

The introduction of Bidirectional Encoder Representations from Transformers (BERT) by Google in 2018 (Devlin et al., 2018 [3]) marked a significant milestone in the field of natural language processing (NLP), enabling state-of-the-art performance across tasks such as classification, question answering, and entity recognition. Leveraging the transformer architecture and contextual word representations, BERT has proven highly effective and versatile. Despite its advantages, deep learning models like BERT often require substantial computational resources and time for training and evaluation, necessitating technologies like NVIDIA's CUDA GPU processing. This complexity raises questions about whether the performance improvements justify the added resource demands compared to traditional models. Studies such as Zhang et al. (2020 [8]) have demonstrated the efficacy of BERT in e-commerce sentiment analysis. Using an Amazon Review dataset with fields like reviewerID, product IDs, reviewText, summaries, and overall ratings, Zhang et al. trained their model on reviewerID and reviewText, achieving an accuracy of 80.1%, a recall score of 0.736, and an F1 score of 0.767. In this study, we aim to achieve findings that align with their results, further validating the robust performance of BERT in sentiment analysis tasks.

Logistic Regression models have also been applied to sentiment analysis using written language to perform classification and predictive tasks. This statistical algorithm uses a sigmoid function to take inputs as independent variables and produces a probability value between 0 and 1, analyzing the relationship between two data factors. For instance, with five categories (1, 2, 3, 4, 5), the probabilities could be set as 0.2, 0.4, 0.6, 0.8, and 1. If the logistic function value for an input is 0.9, it would be classified as a 5. A study conducted by George B. Aliman on sentiment analysis classified tweets in English, Filipino, and Taglish languages, using Twitter's API to gather data. Among SVM, Stochastic Gradient Descent, Naive Bayes, and Logistic Regression, the results showed that Logistic Regression had the highest accuracy and best-fitted algorithm

for predicting mental health crisis tweets, achieving 81% accuracy (Aliman et al., 2022 [1]). This project will employ a similar approach to categorize Amazon product reviews, corresponding to a 5-star rating system.

Random Forest models leverage ensemble learning by combining multiple instances of the same algorithm or different algorithms to enhance predictive power (Bahwari, 2019 [2]). This approach uses a tree-structured classifier where each node is split based on the most effective predictor from a randomly selected subset. Random forests are highly robust and flexible, excelling in modeling non-linear relationships and complex variable interactions. A sentiment analysis study using random forests on tweets from airline services achieved approximately 76% accuracy in classifying sentiments into positive, negative, and neutral categories. This project will apply both logistic and random forest regression models to classify Amazon product reviews, comparing their performance to identify the superior model for our dataset.

Support Vector Machine (SVM) is a supervised learning algorithm that can be used for classification and regression tasks. SVM aims to find the optimal hyperplane that best separates data points belonging to different classes. In the context of sentiment classification, the goal is to model the relationship between the sentiment labels (1-5 stars) and the feature vectors (TF-IDF features of reviews). Unlike LDA, SVM does not assume a particular distribution for the data, making it more flexible in handling real-world data that may not follow specific statistical assumptions. Regarding linear classification, consider a binary classification problem with a dataset where each independent variable is a feature vector and the dependent variable is the class label. The objective of SVM is to find the optimal hyperplane that separates the two classes by maximizing the margin. The hyperplane is represented by:

$$M = \frac{1}{\|\mathbf{w}\|}$$

where w is the weight vector normal to the hyperplane, and b is the bias term. The margin M is defined as the distance from the hyperplane to the nearest data points (support vectors). The margin is given by:

$$\mathbf{w}^T\mathbf{x} + b = 0$$

3 Data

The product reviews used in this thesis are from an e-commerce dataset made available on Kaggle. This dataset provides a comprehensive collection of customer reviews for various products, making it an invaluable resource for understanding consumer behavior and evaluating product performance. Each record in the dataset includes detailed information such as the review text, a summary of the review, the reviewer's name, and the product's Amazon Standard Identification Number (ASIN). Reviews also include additional metadata like the overall rating (on a scale from 1 to 5 stars), the number of helpful votes received, and the time the review was posted, both in Unix timestamp and human-readable formats.

The dataset is particularly rich, as it captures a wide range of variables useful for sentiment analysis. For the purposes of this thesis, the review text and overall product rating are the primary variables used to construct a sentiment classification system. The overall rating is categorized into three sentiment classes: 1-2 stars as negative, 3 stars as neutral, and 4-5 stars as positive. This classification provides a structured way to analyze customer feedback and extract meaningful patterns from the unstructured text data. This subset was chosen to ensure a balanced representation across sentiment classes to the extent possible, aiding in the development of a robust sentiment classification model. This approach allows for efficient processing while maintaining the integrity of the analysis.

The bar chart illustrates the distribution of sentiment ratings across five categories: 5 Stars, 4 Stars, 3 Stars, 2 Stars, and 1 Star. The most frequent sentiment is 5 Stars, with a total of 3,921 occurrences, reflecting a strong positive response. In contrast, the 4 Star rating follows with 527 occurrences, showing a noticeable decline. The 3-star rating is further reduced, with only 244 occurrences, and the 2-star rating has 142 occurrences. The 1-star rating is the least common, with just 80 occurrences, indicating that very

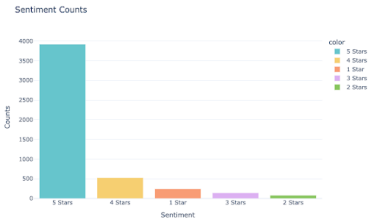

Figure 1: Sentiment Counts of Product Data

Figure 2: Word Clouds of 5 star (Positive) and 1 star (Negative) Product Reviews

few respondents expressed a highly negative sentiment. This pattern suggests a skew toward more positive feedback, with fewer individuals rating lower than 3 Stars.

Word clouds were utilized to filter out positive and negative reviews, extract their cleaned text, and generate a visualize the most common words in those reviews, providing insight into the frequent themes or complaints from customers who gave low ratings. As shown in Figure 2, there were little to no singular words present within the reviews there used by the programs to efficiently discriminate between low-star and high-star reviewed products providence evidence that the reviews' word order, phrase level meaning, dependency parsing, and overall contextual language interpretation were utilized as opposed to pure lexical features.

4 Methodology

4.1 Data Preprocessing

The data preprocessing stage was relatively straightforward. After importing the dataset from Kaggle, we created a text_cleaning function that processed text data by parsing HTML content with BeautifulSoup to extract plain text. We then utilized a regular expression to remove any non-alphanumeric characters except spaces, commands, and apostrophes. The cleaned text was returned, ensuring that the text was standardized and easy to analyze. The text was then converted to numerical features using the TF-IDF function, allowing it to be combined with other numerical variables such as day_diff, helpful_yes, and total_vote.

4.2 Models

4.2.1 BERT

The BERT model requires specific input formatting, using a WordPiece tokenizer to break down words into subunits, creating meaningful text representations (Wu et al,. 2016 [8]). The BERT vocabulary, containing 30,522 tokens, includes subwords or individual characters for unknown words. BERT is pretrained on Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) tasks, which help capture both word- and sentence-level relationships, making it effective for various tasks. For sentiment classification, a feedforward neural network (FFNN) classifier is added on top of BERT, with a softmax activation function

for output probability distributions. During fine-tuning, the Adam optimization algorithm adjusts the model parameters. Fine-tuning was done using Python, PyTorch, and the Transformers library on GPUs via Google Colab due to high computational demands. Hyperparameter tuning was performed by allocating 10% of the dataset for validation and 10% for testing. The optimal batch size was 16, and training stopped after 5 epochs when the model reached its peak performance. The Adam optimizer used a learning rate of 2e-6 to minimize loss and prevent overfitting.

4.2.2 Logistic Regression

The second model implemented was a logistic regression. Two versions of the model were tested: one with reviewerID, reviewText, summary, and overall, and another with six variables, which included the original four plus helpful and day_diff. The reviewerID variable was included solely to link the reviews with the corresponding product owners. The first model focused on looking at the correlation between reviewText and summary data to predict the overall scores, while the second model incorporated additional variables that may assist in score prediction. The dataset, which contains 4,914 data points, was divided into training and testing sets following the 80/20 rule. The classification report and accuracy were outputted, and a confusion matrix was graphed to show the predicted versus actual labels.

4.2.3 Random Forest Model

Similar to the logistic regression, the random forest model used the same 4 variables: reviewerID, reviewText, summary, and overall. It had the same goal as the previous model, and the dataset was split in the same 80/20 rule. We outputted a classification report, accuracy, and a confusion matrix to compare to the logistic regression model.

4.2.4 SVM

After preprocessing the data, the reviews were converted into numerical feature vectors using the TfidfVectorizer from sklearn, applying Term Frequency-Inverse Document Frequency (TF-IDF) to highlight relevant words. The vectorizer was fit on the training dataset and then applied to the validation and test datasets.

For model training, a linear SVM classifier was used with the One-vs-Rest (OvR) strategy to manage the multi-class nature of the problem (ratings 0-4). The SVM was configured with a linear kernel, which effectively separates data by maximizing the margin between classes. The decision_function_shape='ovr' parameter trains a separate classifier for each class to distinguish it from the others. During prediction, the model selects the class with the highest confidence score.

The vectorized training data was fed into the SVM's fit method, where the algorithm minimizes a loss function to achieve a large margin between classes. Given the high-dimensional nature of text data, the linear kernel uses stochastic gradient descent (SGD) to find the best solution efficiently.

5 Results

We observed that our dataset was relatively small and imbalanced, with 3,921 out of 4,914 data points labeled as a 5, indicating that it was highly left-skewed or biased towards positive feedback. To address this imbalance, we tried implementing several techniques including Synthetic Minority Over-sampling (SMOTE), random undersampling, weight adjustments, and regularization. Here are our results.

5.1 BERT

The classification report shows that while the model achieved an overall accuracy of 82%, its performance varied significantly across different classes. It performed well on 5-star reviews, with a precision of 0.95, recall of 0.89, and an F1-score of 0.92, indicating strong accuracy in identifying this class. However, the model struggled with the other ratings, particularly the 1-star, 2-star, and 3-star reviews, which had near-zero precision and recall. The 4-star reviews also showed poor performance with a precision of 0.13 and recall of 0.24. These results highlight a strong bias toward the 5-star class, likely due to its overrepresentation in

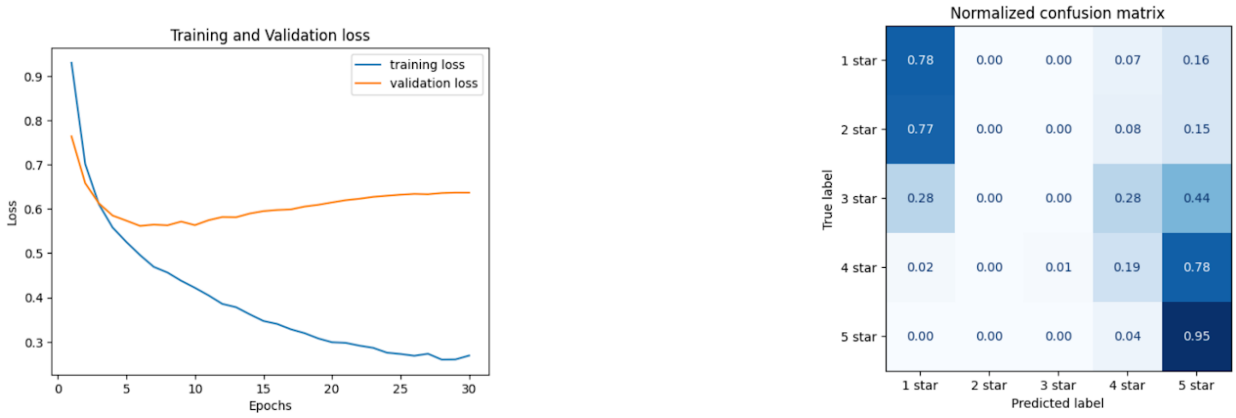

Figure 3: BERT Training and Validation Loss + Confusion Matrix

the dataset. The low macro average precision, recall, and F1-score further reflect this imbalance, while the higher weighted average scores are influenced by the model's success with the dominant 5-star class. One key approach to address this would be class balancing, which could involve oversampling the minority classes or undersampling the majority class to ensure more even representation. However, this would ideally be done with supplementing the dataset to have an overall higher percentage of samples for classes other than 5 stars.

5.2 Logistic Regression

Figure 4 displays the accuracy and classification report. The first logistic regression model achieved a classification accuracy of 82.5%, indicating that the chosen variables had a linear relationship with the "overall" predicted score. However, the precision for labels 2-4 were surprisingly low, and the confusion matrix highlights this skew towards predicting category 5, especially for true reviews that should belong in 1 and 4. We believe that accuracy could improve with more data, as the current model struggled to distinguish subtle differences that might place a review in the 4 category rather than a 5.

Accuracy: 0.8250254323499492

Classification Report:

	precision	recall	f1-score	support
1	0.70	0.57	0.63	49
2	0.00	0.00	0.00	16
3	0.14	0.04	0.06	28
4	0.28	0.05	0.08	106
5	0.85	0.99	0.91	784
accuracy			0.83	983
macro avg	0.39	0.33	0.34	983
weighted avg	0.74	0.83	0.77	983

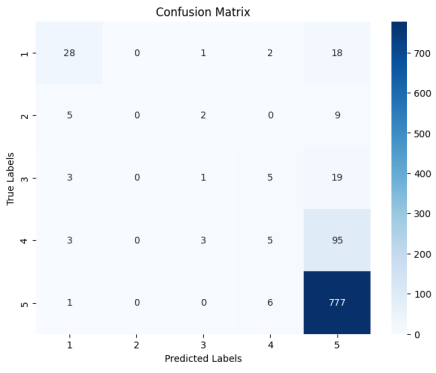

Figure 4: Classification Report and Confusion Matrix for Logistic Regression (4 variables)

Analyzing ten product reviews that incorrectly predicted products into category 5 instead of 4 can further reveal the model's weaknesses. The first review, simply stating "No issues," is extremely straightforward with limited context, making it challenging for the model to accurately distinguish between a 4 and a 5 rating. Additionally, many of these examples were difficult to categorize correctly, even for a human, as they conveyed no clear excitement or emotional response. This subjective nature of online reviews adds complexity, with younger reviewers more likely to use emojis and symbols, potentially influencing the perceived quality

differently than older demographics.

Accuracy: 0.8240081383519837

Classification Report:

	precision	recall	f1-score	support
1	0.69	0.55	0.61	49
2	0.00	0.00	0.00	16
3	0.14	0.04	0.06	28
4	0.28	0.05	0.08	106
5	0.85	0.99	0.91	784
accuracy			0.82	983
macro avg	0.39	0.32	0.33	983
weighted avg	0.74	0.82	0.77	983

Figure 5: 10 Incorrect Product Reviews and Logistic Regression (6 variable) Classification Report

The second logistic regression model showed a slight decrease in classification accuracy to 82.4%. This reduction is likely due to overfitting, as the added variables did not have a linear correlation with reviewText and summary, introducing noise and complicating the model's ability to accurately predict the overall scores.

5.3 Random Forest Model

The figure displays the accuracy and the classification report for the Random Forest model. To prevent overfitting, the model was restricted to using reviewerID, reviewerText, summary, and overall features. The classification accuracy was 80.2%, which was slightly lower than that of the logistic regression model. Additionally, the confusion matrix showed an increase in incorrect predictions for the 1 and 4 categories. Although Random Forest is generally more flexible and capable of capturing complex interactions in data due to its ensemble nature, it can suffer from significant class imbalance. Each decision tree may be biased towards the majority class, leading to poor performance on the minority class, as seen in categories 1 and 4 in our dataset. While the logistic regression model performed slightly better in this case, adding more data could help improve the Random Forest's performance. We believe a balanced dataset could reveal more promising results for the Random Forest model in future comparisons.

Accuracy: 0.8016276703967447

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.10	0.19	49
2	0.00	0.00	0.00	16
3	0.00	0.00	0.00	28
4	0.00	0.00	0.00	106
5	0.80	1.00	0.89	784
accuracy			0.80	983
macro avg	0.36	0.22	0.21	983
weighted avg	0.69	0.80	0.72	983

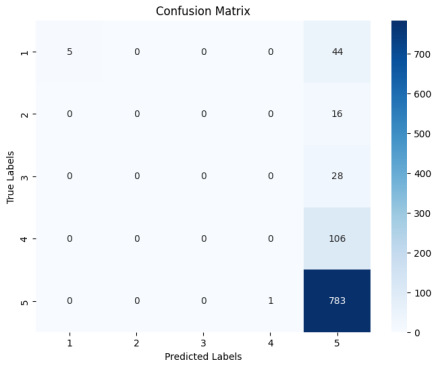

Figure 6: Classification Report and Confusion Matrix for Random Forest (4 variables)

5.4 SVM

When comparing the two SVM classification reports, we can observe that both models achieve the same overall accuracy of 83%, but the inclusion of additional features, such as the day difference between review

and purchase, and the date of review, does not lead to a significant improvement in performance across the different classes.

In the first model, the recall for 5-star reviews is perfect (1.00), reflecting the model's bias toward the dominant class. However, the model struggles with other classes, especially 1-star, 2-star, 3-star, and 4-star reviews, which have low precision and recall. The 1-star class shows moderate performance (precision 0.67, recall 0.43), while the 2-star and 3-star classes perform poorly with precision and recall of 0.00. The second model, with additional features like day difference and review date, shows a similar pattern. The precision for 1-star reviews improves to 0.74, but recall remains unchanged at 0.43. The 3-star class has perfect precision (1.00) but very low recall (0.04), indicating the model can classify some reviews but misses most of them. The 2-star and 4-star classes still perform poorly, suggesting the additional features had limited impact.

Despite the added features, the overall improvements seem minimal. The macro average precision and recall for both models remain low, particularly for the minority classes, indicating that the class imbalance continues to heavily influence model performance. This also suggests that the additional features might not be providing enough discriminative power to handle the imbalanced nature of the dataset effectively.

Permutation importance can help identify which features most impact the SVM model's performance. With multiple customer-related columns, it could highlight valuable features for distinguishing sentiment beyond just the text data.

Results for SVC(kernel=linear)
Training time: 4.024073s; Prediction time: 1.092658s

	precision	recall	f1-score	support
1.0	0.67	0.43	0.53	46
2.0	0.00	0.00	0.00	13
3.0	0.00	0.00	0.00	26
4.0	0.00	0.00	0.00	106
5.0	0.83	1.00	0.91	792
accuracy			0.83	983
macro avg	0.30	0.29	0.29	983
weighted avg	0.70	0.83	0.76	983

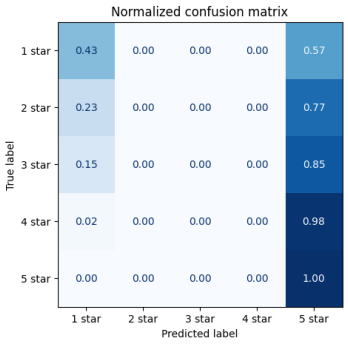

Figure 7: Classification Report and Confusion Matrix for SVM without additional features

Classification Report:

	precision	recall	f1-score	support
1.0	0.74	0.43	0.55	46
2.0	0.00	0.00	0.00	13
3.0	1.00	0.04	0.07	26
4.0	0.00	0.00	0.00	106
5.0	0.83	1.00	0.91	792
accuracy			0.83	983
macro avg	0.51	0.29	0.31	983
weighted avg	0.73	0.83	0.76	983

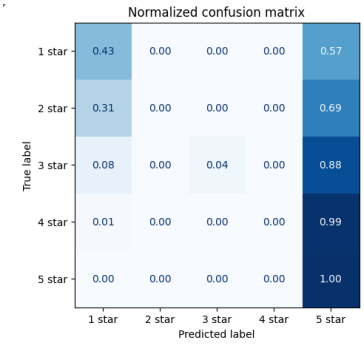

Figure 8: Classification Report and Confusion Matrix for SVM with additional features

6 Alternatives

To improve the performance of non-BERT models, several key enhancements were made to data preprocessing and feature engineering compared to previous approaches. Besides vectorizing the text data, additional features were extracted from the dataset, including the reviewTime and day_diff columns. The reviewTime

column was converted to a Unix timestamp to provide a numeric representation of when each review was written, and the day_diff feature measures the difference in days between the review date and a reference point to capture temporal trends in review patterns.

To ensure these additional features were appropriately scaled, the StandardScaler standardized them to have a mean of 0 and a standard deviation of 1. These time-based features were then converted into sparse matrices using csr_matrix to optimize memory usage. The TF-IDF feature vectors were combined with the time-based features using the hstack function from scipy.sparse, creating a unified feature matrix.

Given the heavy skew towards 5-star reviews, which had significantly more examples than other classes, the dataset was balanced using undersampling to ensure each class had an equal number of samples in training, validation, and test sets. However, this balancing technique led to underperformance across all classes, including the 5-star class. As a result, this approach was excluded from the final training of the models.

7 Conclusion

This project centered on sentiment analysis classification of Amazon product reviews, with a focus on training the BERT bidirectional transformer model. The performance of BERT was compared to other natural language processing models, including logistic regression, random forest, and support vector machines. As anticipated, BERT outperformed the other models, achieving an overall accuracy of 82%, a precision of 0.95, and a recall score of 0.89. These results demonstrate BERT's effectiveness in capturing the relationships between review content and classification outcomes.

The dataset was skewed towards 5-star reviews, creating an imbalance that likely affected classification performance. While the model accurately predicted positive sentiments, it struggled with underrepresented lower ratings. BERT outperformed SVM, random forest, and logistic regression due to its ability to understand contextual relationships in the reviews through its transformer architecture. Unlike traditional models, which rely on predefined features, BERT leverages pre-trained knowledge and attention mechanisms, enabling it to capture language nuances and make more accurate predictions, especially where context is key.

References

[1] Aliman, George B., et al. "Sentiment Analysis using Logistic Regression." Journal of Computational Innovations and Engineering Applications, vol. 2, no. 1, 2022, pp 35-40.

[2] Bahrawi, Kominfo. "Sentiment Analysis using Random Forest Algorithm–Online Social Media Based." Journal of Information Technology and its Utilization, vol. 2, no. 2, Dec. 2019, pp 29-33.

[3] Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805. https://arxiv.org/abs/1810.04805

[4] Liu, Bing. "Sentiment Analysis and Opinion Mining". Morgan and Claypool Publishers (2012).

[5] Lu, G., Zhao, X., Yin, J., Yang, W., and Li, B. "Multi-task learning using variational autoencoder for sentiment classification." Pattern recognition letters 132:115–122, 2020.

[6] Kang, Min, et al. "A study on the influence of online reviews of new products on consumers' purchase decisions: An empirical study on JD. com." Frontiers in Psychology 13 (2022): 983060.

[7] Kang, Min, et al. "A study on the influence of online reviews of new products on consumers' purchase decisions: An empirical study on JD. com." Frontiers in Psychology 13 (2022): 983060

[8] Zhang, Shaozhong et al, "A Multiclassification Model of Sentiment for E-Commerce Reviews," in IEEE Access, vol. 8, pp. 189513-189526, 2020, doi: 10.1109/ACCESS.2020.3031588