Rating Predictions from reviews given to products in online markets

Dmitry Lukyanov
Applied Data Science
CPSC-6300, Spring 2023
(SVM,RNN, EDA, Report)
dlukyan@clemson.edu

Sejal Bansal
Applied Data Science
CPSC-6300, Spring 2023
(CNN, EDA, Report)
sejalb@g.clemson.edu

Adithya Ravi
Applied Data Science
CPSC-6300, Spring 2023
(LSTM, EDA, Report)
aravi@g.clemson.edu

Shareef Shaik
Applied Data Science
CPSC-6300, Spring 2023
(BERT, EDA, Report)
shaik@clemson.edu

Abstract

Fine-grained aspect-based analysis for product reviews is a challenging task in natural language processing. In this research, we aimed to develop and compare a set of models that would effectively perform this analysis on a given dataset. However, we encountered significant challenges with the quality of the data, leading us to make several adjustments to the dataset.

Despite our best efforts, we were unable to create a model that performed as well as we had hoped. Nevertheless, some of the models displayed accuracy that is significantly higher than a baseline. We discuss the implications of our findings and suggest directions for future research in this area.

Keywords: Convolutional Neural Network (CNN), Recuurent Neural Netowrks (RNN), Long Short Term Memory (LSTM), Bidirectional Encoder Representations from Transformers (BERT), SVM, Regular Expression, Receiver Operating Characteristic(ROC), Confusion Matrix, ChatGPT.

Source Code: ***Github Link***

1 Introduction

The explosion of online shopping has created an unprecedented amount of product reviews, with millions of consumers sharing their experiences and opinions on various products. However, analyzing this massive amount of data is a daunting task, and the ability to extract valuable insights from it is critical for businesses to improve their products and services. Fine-grained aspect-based analysis of product reviews has emerged as a crucial task in natural language processing and machine learning, as it allows for a more nuanced understanding of the various aspects that consumers mention in their reviews. The main question that our project seeks to answer is can we develop an effective model for fine-grained aspect-based analysis of product reviews, and what are the challenges that we might face in doing so? We aim to explore this question by utilizing a variety of machine learning techniques and applying them to a dataset of product reviews.

The motivation for our project question is twofold. First, the ability to perform fine-grained aspect-based analysis of product reviews has numerous practical applications, such as improving product design, identifying customer needs, and enhancing customer satisfaction. Second, the challenges that we face in developing an effective model for this task highlight the importance of high-quality data and the need for further research in this area. By addressing these challenges, we can develop more accurate and effective models that can provide valuable insights into consumer preferences and behaviors.

In this report, we will discuss the methods we used to tackle the challenge of fine-grained aspect-based analysis of product reviews. We will also present the results of our experiments and the insights we gained from them.What can be learned from our project is the importance of addressing the challenges in fine-grained aspect-based analysis of product reviews. While we were unable to develop an effective model for this task, our research provides valuable insights into the difficulties and complexities involved.

2 EDA

Our unit of analysis for the dataset is at the review level. Each row in the dataset represents a single review, with features such as the product category, review text, rating and so on. Therefore, the analysis performed on this dataset focuses on the characteristics and distribution of individual reviews, rather than on the products or reviewers themselves.There are 3781 unique observations and since all reviews are generated by Chat-GPT, we are not sure of the time period.

2.1 Dataset

During EDA it was discovered that the dataset is

- Highly unbalanced in ratings

- Highly unbalanced in aspects presented of reviews

To address the issue it was decided to generate an artificial data set of reviews with ChatGPT API, using gpt-3.5-turbo model. In order to reflect the variety in possible aspects combinations in real reviews, the next distribution was designed: single-aspect reviews (1 aspect) = 220 items each aspect type = 55 items multi-aspect reviews (3 aspects) = 60 items each aspect of reviews is missing = 220 items multi-aspect reviews (4 aspects) = 900 items In sum - 1340 items. For each subset there is approximately equal distribution among rates from 1 to 5. The final set contains approximately equal distribution among rates from 0 to 5 where 0 marks missing aspects.

2.2 Data Cleaning

Data cleaning is an important step in any data analysis project to ensure that the data is accurate, complete, and consistent. For checkpoint 1 data set we performed data cleaning steps like removing null values and duplicates, getting rid of punctuation marks and special characters, lowercase conversion, etc. But given that the data set was highly imbalanced, we then generated our data set using chatGPT. The following steps are included in the data cleaning procedure for our data set.

1. HTML tags are removed: HTML tags are typically used to format text in web pages. However, in the context of text analysis, they are not relevant and can cause issues when parsing the text. Therefore, it is necessary to remove HTML tags from the text data before further processing.
2. Emails are removed: Emails are typically irrelevant for text analysis and can cause issues when parsing text. Therefore, it is necessary to remove email addresses from the text data.
3. URLs are removed: URLs are also typically irrelevant for text analysis and can cause issues when parsing text. Therefore, it is necessary to remove URLs from the text data.
4. Accented letters are replaced with standard versions: Accented letters can cause issues when processing text data, as they may not be recognized by some natural language processing tools. Therefore, it is necessary to replace accented letters with their standard versions.
5. Emojis are removed: Emojis are typically irrelevant for text analysis and can cause issues when parsing text. Therefore, it is necessary to remove emojis from the text data.
6. Special symbols are removed: Special symbols such as hashtags, currency symbols, and mathematical symbols are typically irrelevant for text analysis and can cause issues when parsing text. Therefore, it is necessary to remove special symbols from the text data.
7. Excessive spaces are removed: Excessive spaces can cause issues when processing text data, as they may lead to false word counts or other parsing issues. Therefore, it is necessary to remove excessive spaces from the text data.
8. Words contractions are replaced with full forms: Contractions such as "don't" and "can't" can cause issues when processing text data, as they may be interpreted as separate words. Therefore, it is necessary to replace contractions with their full forms.
9. Grammar is fixed: Grammatical errors can cause issues when processing text data, as they may lead to false word counts or other parsing issues. Therefore, it is necessary to fix any grammatical errors in the text data.
10. Letters are lowercased: Lowercasing all letters in the text data can help to standardize the data and make it easier to process. Punctuation is removed: Punctuation such as periods, commas, and exclamation marks are typically irrelevant for text analysis and can cause issues when parsing text. Therefore, it is necessary to remove punctuation from the text data.
11. Words are lemmatized: Lemmatization involves reducing words to their base form (e.g., "running" -> "run"). This can help to standardize the data and make it easier to process.
12. Stopwords are removed: Stopwords are common words that are typically irrelevant for text analysis (e.g., "the", "a", "and"). Removing stopwords can help to reduce noise in the data and make it easier to analyze.
13. Titles and texts were merged: In some cases, the title and text of a review may be analyzed separately. However, in this project, the title and text were merged into a single block of text for analysis.
14. Reviews were vectorized with TF-IDF vectorizer: The TF-IDF (term frequency-inverse document frequency) vectorizer is a common method for converting text data into a numerical format that can be used for analysis. The vectorizer assigns a weight to each word based on its frequency in the document and its rarity in the overall corpus.

However, due to the imperfection of ChatGPT it was impossible to achieve stable results for the whole dataset. E.g., quite regularly when the model was told to generate a review with the product aspect for a product that would be rated as 4 out of 5, it generated a review that didn't contain any flaws for the product and contained statements that the product is perfect. Thus, while the dataset became more balanced, such irregularities affected the model performance negatively. The model GPT-4 was tried, but produced reviews with the same issues.

As due to the training approach that was used for training and that is described below, it was not possible to guarantee that there was no data leakage, for the validation 444 new reviews with the same distribution were generated.

The below are the visualizations developed using the dataset. As the input of the model is text and output is numerical it is very difficult to develop a visualization against the input vs output.

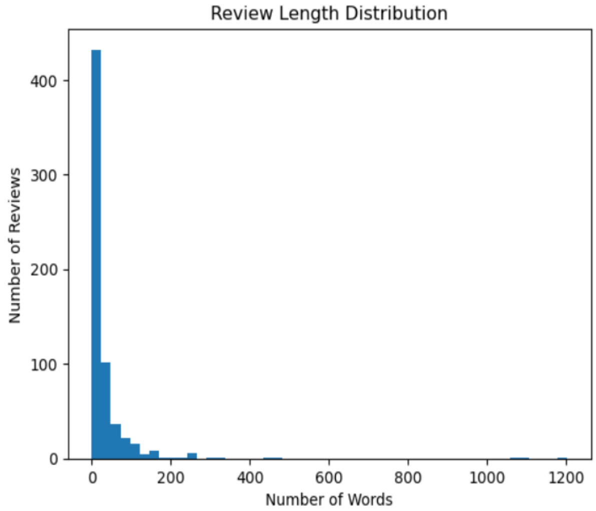


Figure 1: Review Length Distribution

The chart displays the number of reviews on the y-axis and the length of the reviews (in terms of the number of words) on the x-axis. The histogram is divided into bins, where each bin represents a range of review lengths. The height of each bar represents the number of reviews that fall within that particular range of review lengths.

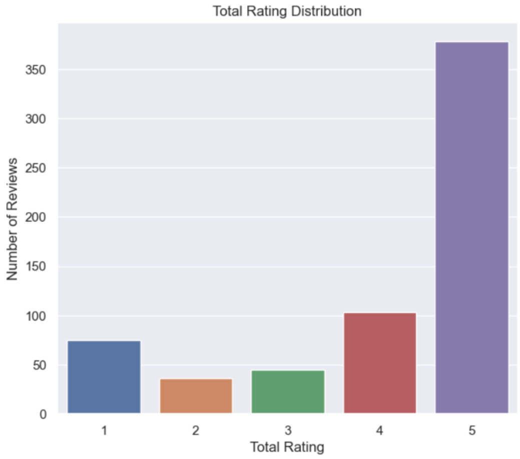


Figure 2: Total Rating Distribution

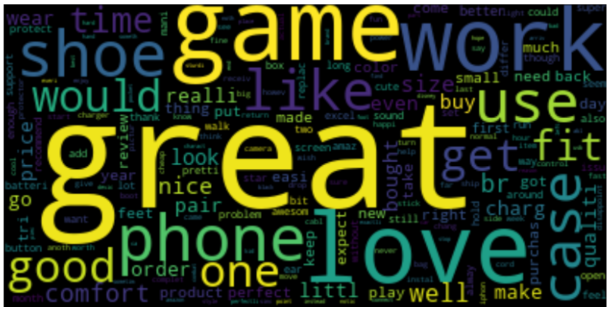


Figure 3: Word Cloud of Reviews

Considering the nature of the project and having features of the dataset, the most reasonable choice of predictors will include in order of importance review_body and review_title.

3 Models

Several options for evaluation of the model's performance for classification were considered:

- AUC-ROC was rejected as it's beneficial for imbalanced datasets that is not the case after introducing the artificially generated dataset.
- Precision and recall and consequently F1-score were rejected as the main interesting metric for us is a number of correctly classified ratings in respect to the real ones.

Thus, accuracy was chosen for model performance evaluation. But, as we train and use separate models for each aspect instead of an multi output-multilevel model, the final metric is the average of models' accuracies for aspects.

3.1 Model 1 (1D CNN)

Model Choice

While convolutional neural networks were originally developed for image recognition tasks, they can also be applied to natural language processing tasks, including text classification. CNNs are particularly useful for text classification because they can extract important features from text data. CNNs are capable of learning local patterns in a text input by using a sliding window over the text to extract n-grams (consecutive sequences of n words) that are relevant to the classification task. These n-grams are then processed through a series of convolutional layers, which extract increasingly complex patterns by applying a set of learned filters to the input. This process results in a feature map that captures the relevant features in the input.

Model training, evaluation and prediction

For each aspect a separate model was trained with hyperparameters tuning with random grid search, 20 epochs and a stop early callback with patience equals to 5. Each time the test set was used for model performance evaluation. In that way the optimal model was trained for each aspect. The mean accuracy was calculated.

Product	Delivery	Seller	Market Place	Overall
14.92%	17.16%	20.90%	14.55%	16.88%

Table 1. Accuracy on the test set

Product	Delivery	Seller	Market Place	Overall
17.79%	19.82%	17.57%	16.89%	18.02%

Table 2. Accuracy on the unseen data

3.2 Model 2 (RNN / LSTM)

Model Choice

Recurrent Neural Networks (RNNs), and specifically Long Short-Term Memory (LSTM) networks, are commonly used for text classification tasks because they are particularly good at processing sequential data, such as text. In contrast to CNNs, RNNs are able to maintain a "memory" of previous inputs, which allows them to capture the temporal dependencies in sequential data. This makes them particularly useful for tasks such as sentiment analysis or language modeling, where the meaning of a word or phrase can be strongly influenced by the context in which it appears.

Model training, evaluation and prediction

For each aspect a separate model was trained with hyperparameters tuning with random grid search, 20 epochs and a stop early callback with patience equals to 5. Each time the test set was used for model performance evaluation. In that way the optimal model was trained for each aspect. The mean accuracy was calculated. While the achieved accuracy for all aspects on the training was comparable to SVM's accuracy from the previous checkpoints and was about 40%, the accuracy for the test set and on the unseen data was no different from a random distribution, considering the fact that there are 6 classes that would give approximately 16.67% accuracy.

Product	Delivery	Seller	Market Place	Overall
13.81%	16.42%	15.30%	16.79%	15.58%

Table 3. Accuracy on the test set

Product	Delivery	Seller	Market Place	Overall
17.12%	17.12%	16.22%	16.44%	16.73%

Table 4. Accuracy on the unseen data

3.3 Model 3 (BERT)

Model Choice

One of the main advantages of using BERT for text classification is that it can take into account the entire context of the input text, rather than just individual words or phrases. This means that it can capture the relationships between words and the nuances of language that are important for accurate classification.

Model training, evaluation and prediction

For enabling BERT we split the reviews into different aspects using regular expressions and several keywords. We believe the splitting accuracy can be increased with more advanced techniques. After processing the data, every aspect is evaluated by the pre-trained BERT model. As the pre-trained model was used, it was possible to directly start predicting with reviews after tokenizing the reviews without additional training. As the aspects are split, it was possible to remove the samples with missing aspects from the aspect-specific predictions, increasing accuracy. The described approach let us achieve approximately 35% accuracy.

Product	Delivery	Seller	Market Place	Overall
13.81%	16.42%	15.30%	16.79%	15.58%

Table 5. Accuracy on the test set

From the confusion matrix and accuracy table we can infer that the model works better than the built CNN and RNN models, but still under performs and cannot be used in production. As we can see, the prediction is diluted around the main diagonal.

The confusion matrices for BERT:

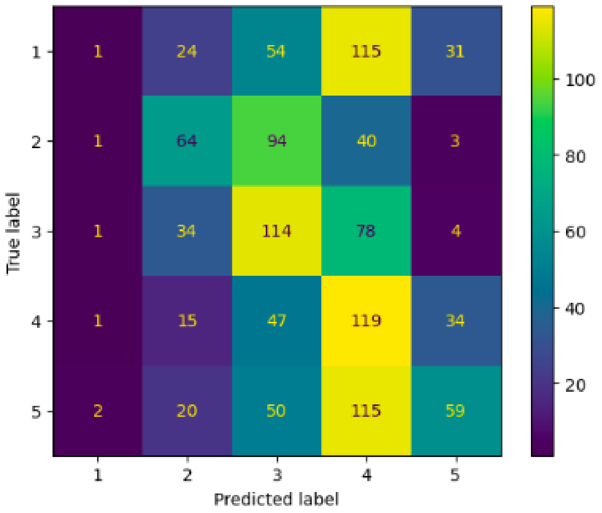


Figure 4: Confusion matrix for BERT (product)

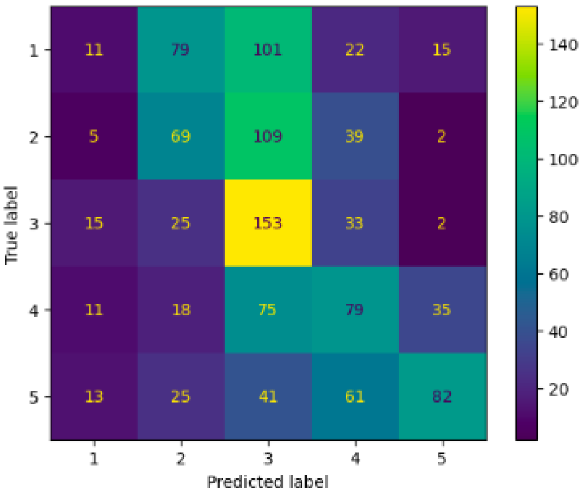


Figure 5: Confusion matrix for BERT (delivery)

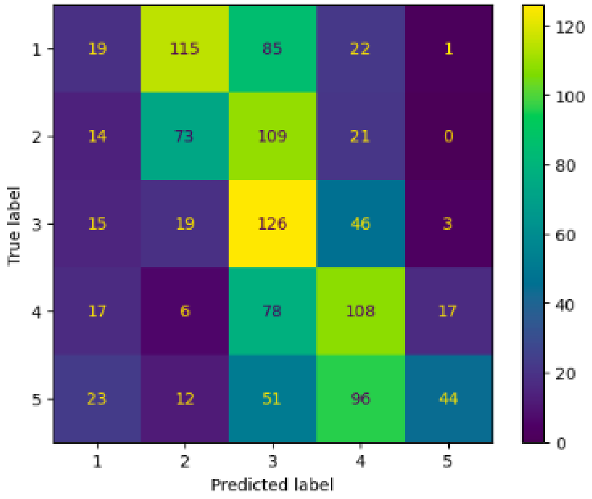


Figure 6: Confusion matrix for BERT (Market Place)

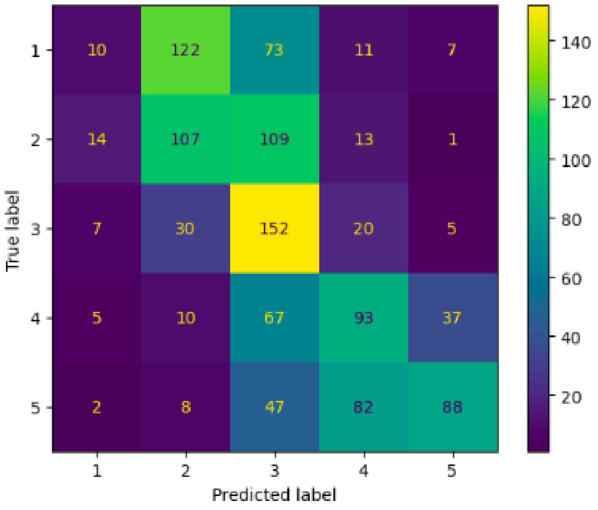


Figure 7: Confusion matrix for BERT (seller)

3.4 Model 4 (SVM)

However, after BERT with aspect splitting achieved the results that are slightly worse than results of SVM without aspect splitting, it was decided to train and evaluate SVM with aspect splitting.

The volume of reviews was split into training and test sets several times in order to avoid a potential situation where a specific split affects model training. For each split for each aspect a separate model was trained with hyperparameters tuning with random grid search and K-fold cross-validation with 5 folds. Each time the test set was used for model performance evaluation. In that way for each split the optimal model was trained for each aspect. For each aspect the mean accuracy was calculated for all trained models and all aspects.

The achieved mean accuracy for all trained models for the test set that is overall higher by 3%+ than for SVM without aspect splitting.

Product	Delivery	Seller	Market Place	Overall
34.12%	41.77%	44.70%	43.36%	40.99%

Table 6. Accuracy on the test set

As the accuracy is calculated as a mean for multiple models, it's not possible to build a confusion matrix for this specific case, and it will be addressed further in the document.

As due to the approach that was used for training it's not possible to guarantee there is no data leakage for a specific model if a random sample of data is taken for validation and prediction after training is completed, 444 were generated separately as a validation set and included into fitting of the vectorizer that was used for reviews encoding, but not used for model training, were used for model validation.

To evaluate and validate models, the noted validation set was used. All trained models were combined into an ensemble with equal weights that allowed to increase the overall accuracy. The achieved accuracy for the ensemble on the validation set that is overall higher by 1%+ than for SVM without aspect splitting.

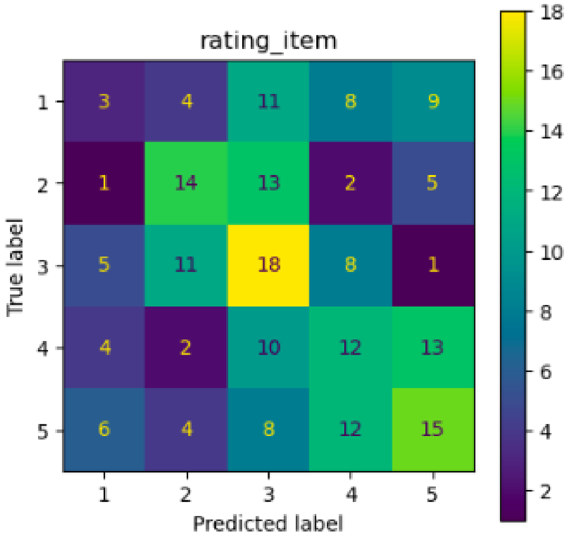


Figure 8: Confusion matrix for SVM(product)

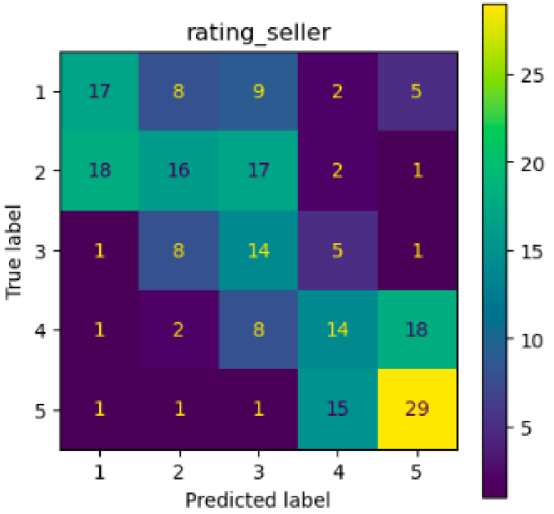


Figure 9: Confusion matrix for SVM(seller)

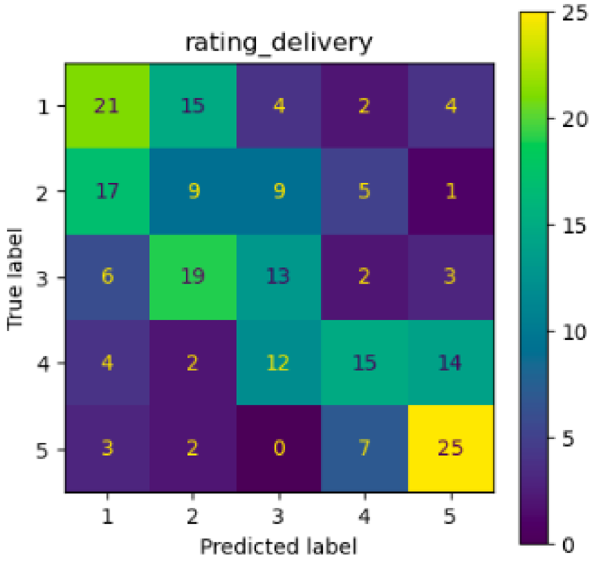


Figure 10: Confusion matrix for SVM(delivery)

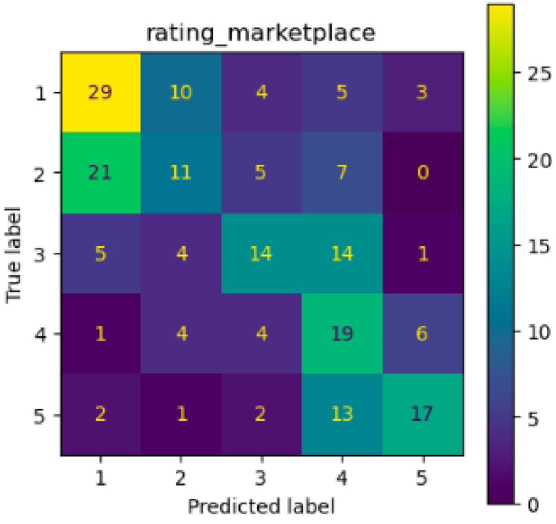


Figure 11: Confusion matrix for SVM(Marketplace)

Product	Delivery	Seller	Market Place	Overall
35.71%	41.67%	44.39%	40.90%	40.47%

Table 7. Accuracy on the test set

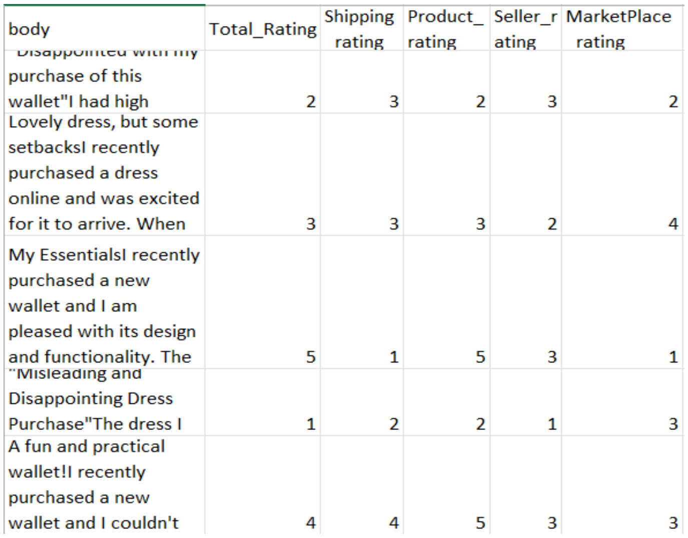
As in one case the accuracy is calculated as a mean for a set of models and in another using the ensemble, it's not possible to evaluate each specific model in detail. To address this issue the best models per aspect were taken. The achieved accuracyfor such models on the validation set that is overall higher by less than 1% than for SVM without aspect splitting.

For both the ensemble and the best models from the confusion matrices it's visible that the models are capable of grasping the general attitude of the reviews, but struggles to recognize a specific rating.

Product	Delivery	Seller	Market Place	Overall
31.16%	38.79%	42.06%	44.56%	39.14%

Table 8. Accuracy on the test set

Below is the snapshot for the predictions for the best model: The snapshot consists of review_text which is labelled as body, the whole review is not clearly visible in the image as the review body is very long, it also consists of total_rating of the product, the shipping_rating, product_rating, seller_rating and marketplace_rating.



body	Total_Rating	Shipping rating	Product_ rating	Seller_r ating	MarketPlace rating
Disappointed with my purchase of this wallet"I had high	2	3	2	3	2
Lovely dress, but some setbacksI recently purchased a dress online and was excited for it to arrive. When	3	3	3	2	4
My EssentialsI recently purchased a new wallet and I am pleased with its design and functionality. The	5	1	5	3	1
"Misleading and Disappointing Dress Purchase"The dress I	1	2	2	1	3
A fun and practical wallet!I recently purchased a new wallet and I couldn't	4	4	5	3	3

Figure 12: Prediction with SVM for 5 different cases (Best Model)

4 Summary & Conclusion

While the low accuracy of CNN and RNN models in comparison to the SVM model are not expected, it's possible to assume that the models were not complex enough to extract the features and generalize. However, due to limited time it was not possible to experiment with models further, especially considering that even the proposed RNN model was taking about 24 hours for training and tuning, while the SVM model provided significantly better results after just 25 minutes.

Expectation for the pre-trained BERT model was even higher than for CNN and RNN models, especially considering that aspect splitting was used. However, the model provided results that are slightly worse that the SVM model without aspect splitting.

Considering that the SVM model with aspect splitting provided results that are just slightly better than the results of the SVM model without aspect splitting and similar performance of the pre-trained BERT model, it's possible to conclude that all three models met a soft cap in accuracy for this dataset and any improvement will be insignificant. As it was mentioned in the previous checkpoint, due to the imperfection of ChatGPT it was impossible to achieve stable results during the dataset generation. E.g., quite regularly when the model was told to generate a review with the product aspect for a product that would be rated as 4 out of 5, it generated a review that did not contain any flaws for the product and contained statements that the product is perfect. Thus, such irregularities affected the model performance negatively, and made it impossible to achieve a good accuracy regardless of the model.

We have devised a method for predicting ratings for the given product in the online markets, we expect this work to help sellers to understand what to improve or how to compose offers to increase the sales of their product. We can improve the accuracy of the models by getting a new dataset. Some models which we developed have a decent accuracy which is more than the random guess.

Due to the imperfection of ChatGPT it was impossible to achieve stable results during the dataset generation. E.g., quite regularly when the model was told to generate a review with the product aspect for a product that would be rated as 4 out of 5, it generated a review that didn't contain any flaws for the product and contained statements that the product is perfect. Thus, such irregularities affected the model performance negatively, and made it impossible to achieve a good accuracy regardless of the model.

5 Future Works

As it was mentioned in the previous section, it seems it is not possible to improve the results with the same dataset, so the most important step would be replacing it with a better one, that would be collected from the real data instead of relying on the imperfect reviews generators. Without this improvement, it's quite probable that any other improvement won't change the situation.

Another way for improvement would be replacing regex-based aspect splitting with more advanced techniques, e.g., capsule networks.