RATING PREDICTIONS FROM REVIEWS GIVEN TO PRODUCTS IN ONLINE MARKETS

CPSC-6300 Applied Data Science

Spring 2023

Checkpoint 3

Dmitry Lukyanov
Sejal Bansal
Adithya Ravi
Shareef Shaik

Codebase

https://github.com/dlcpsc6300/project

Dataset

The same dataset that was used for checkpoint 2 was used. It has 1340 items that are distributed in the next way:

- 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

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.

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 a separate dataset of 444 new reviews with the same distribution was used.

Model performance evaluation

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.

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.

While the achieved accuracy for all aspects on the training set 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**	**Marketplace**	**Overall**
14.92%	17.16%	20.90%	14.55%	16.88%

Table 1. Accuracy on the test set

Product	**Delivery**	**Seller**	**Marketplace**	**Overall**
17.79%	19.82%	17.57%	16.89%	18.02%

Table 2. Accuracy on the unseen data

The confusion matrices for the unseen data are

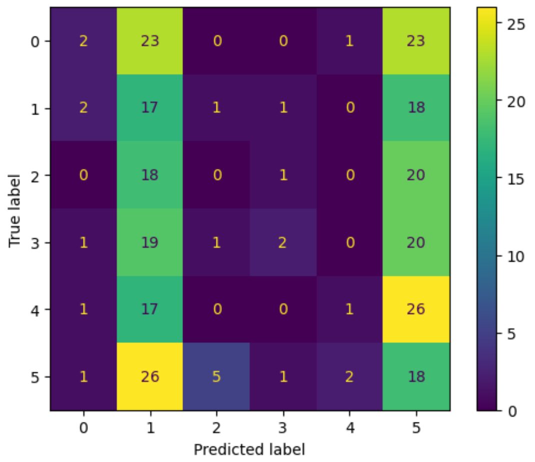


Figure 1. Confusion matrix for CNN (product)

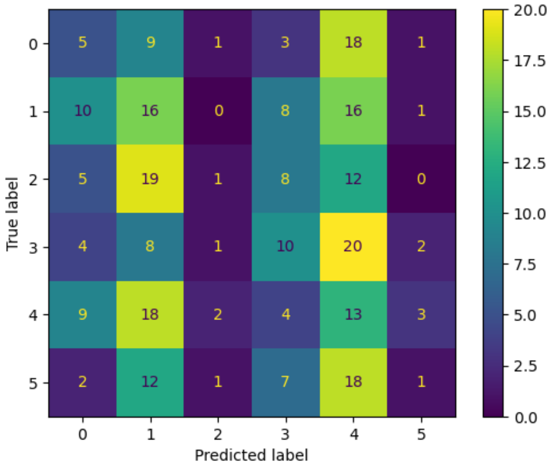


Figure 2. Confusion matrix for CNN (delivery)

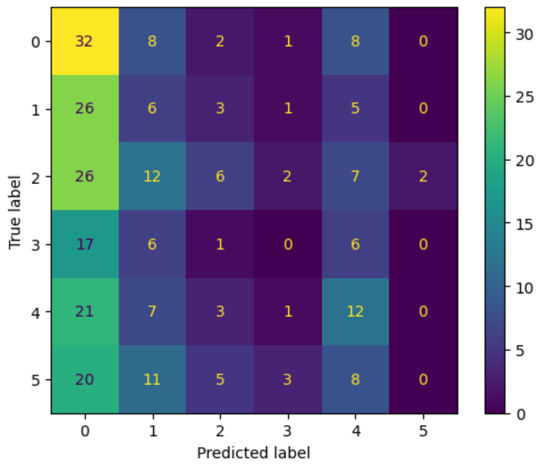


Figure 3. Confusion matrix for CNN (seller)

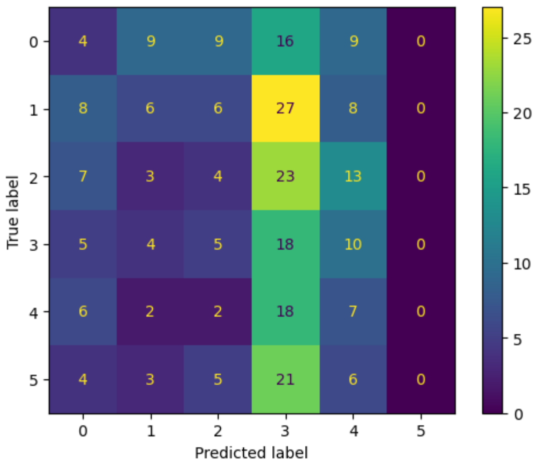
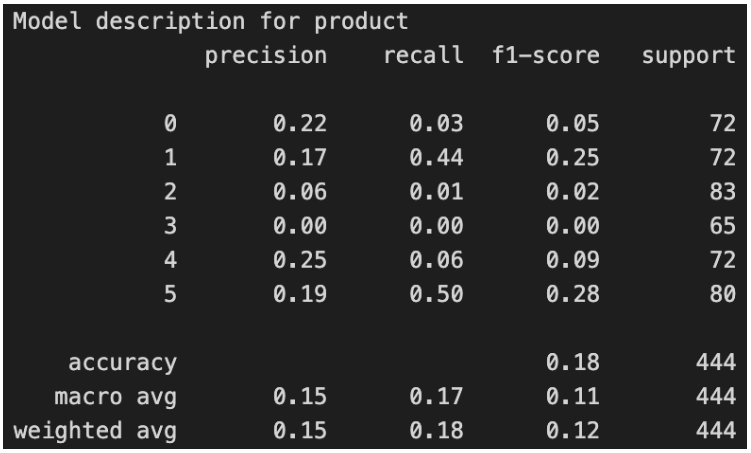
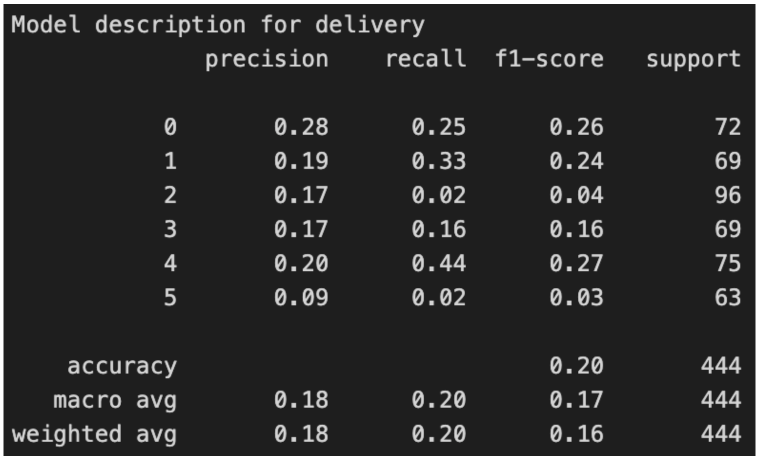


Figure 4. Confusion matrix for CNN (marketplace)

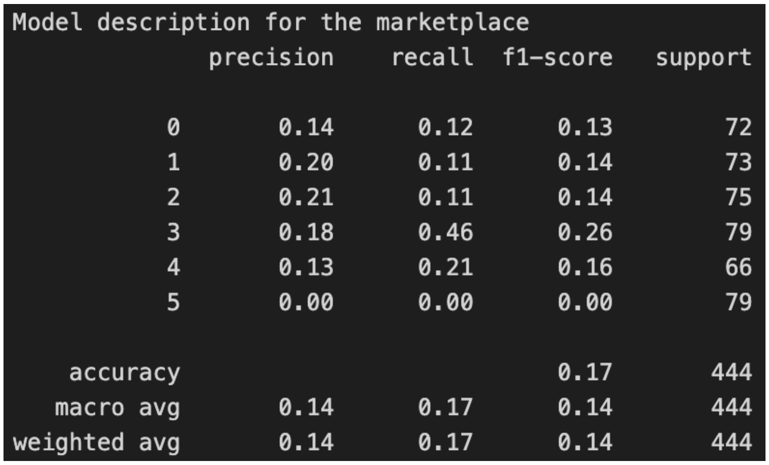
From the confusion matrices it’s visible that the models were not capable of generalizing. There are the models descriptions for the best models per aspect



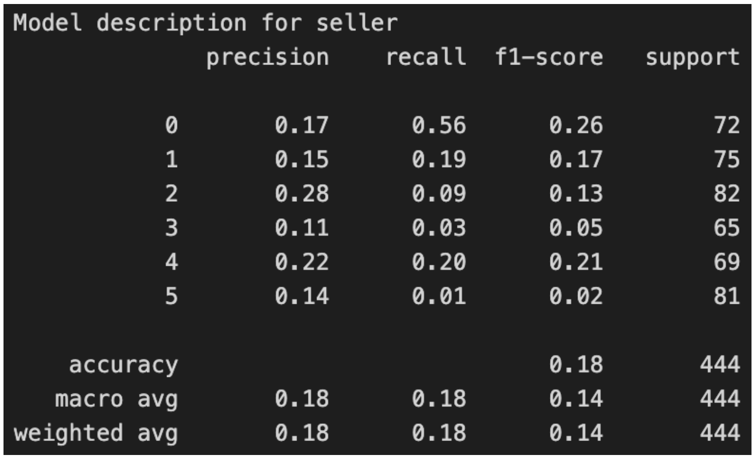
```
Model description for product
              precision    recall  f1-score   support

           0       0.22      0.03      0.05        72
           1       0.17      0.44      0.25        72
           2       0.06      0.01      0.02        83
           3       0.00      0.00      0.00        65
           4       0.25      0.06      0.09        72
           5       0.19      0.50      0.28        80

    accuracy                           0.18       444
   macro avg       0.15      0.17      0.11       444
weighted avg       0.15      0.18      0.12       444
```



```
Model description for delivery
              precision    recall  f1-score   support

           0       0.28      0.25      0.26        72
           1       0.19      0.33      0.24        69
           2       0.17      0.02      0.04        96
           3       0.17      0.16      0.16        69
           4       0.20      0.44      0.27        75
           5       0.09      0.02      0.03        63

    accuracy                           0.20       444
   macro avg       0.18      0.20      0.17       444
weighted avg       0.18      0.20      0.16       444
```



```
Model description for the marketplace
              precision    recall  f1-score   support

           0       0.14      0.12      0.13        72
           1       0.20      0.11      0.14        73
           2       0.21      0.11      0.14        75
           3       0.18      0.46      0.26        79
           4       0.13      0.21      0.16        66
           5       0.00      0.00      0.00        79

    accuracy                           0.17       444
   macro avg       0.14      0.17      0.14       444
weighted avg       0.14      0.17      0.14       444
```



```
Model description for seller
              precision    recall  f1-score   support

           0       0.17      0.56      0.26        72
           1       0.15      0.19      0.17        75
           2       0.28      0.09      0.13        82
           3       0.11      0.03      0.05        65
           4       0.22      0.20      0.21        69
           5       0.14      0.01      0.02        81

    accuracy                           0.18       444
   macro avg       0.18      0.18      0.14       444
weighted avg       0.18      0.18      0.14       444
```

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	Marketplace	Overall
13.81%	16.42%	15.30%	16.79%	15.58%

Table 3. Accuracy on the test set

Product	Delivery	Seller	Marketplace	Overall
17.12%	17.12%	16.22%	16.44%	16.73%

Table 4. Accuracy on the unseen data

The confusion matrices for the unseen data are

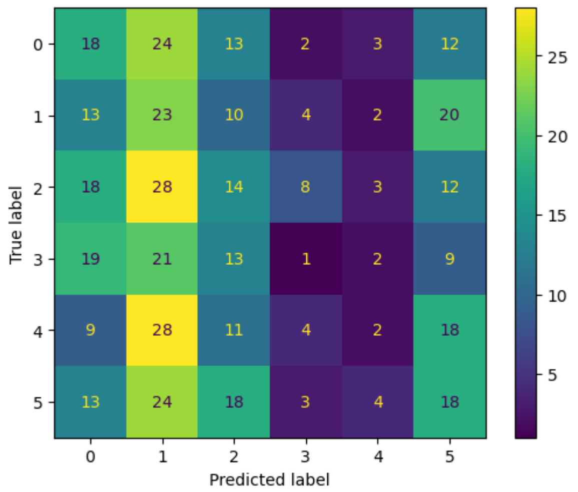


Figure 5. Confusion matrix for RNN (product)

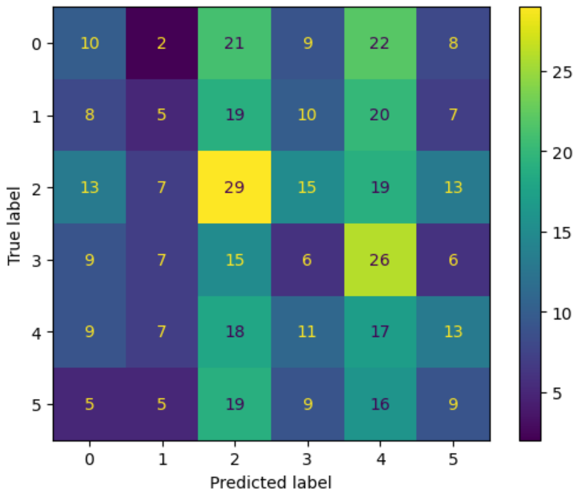


Figure 6. Confusion matrix for RNN (delivery)

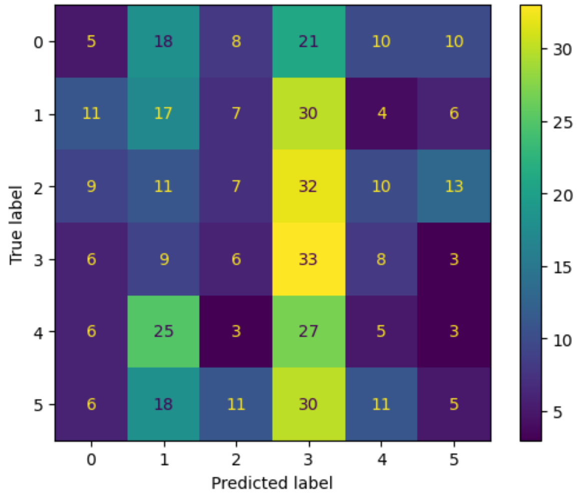


Figure 7. Confusion matrix for RNN (seller)

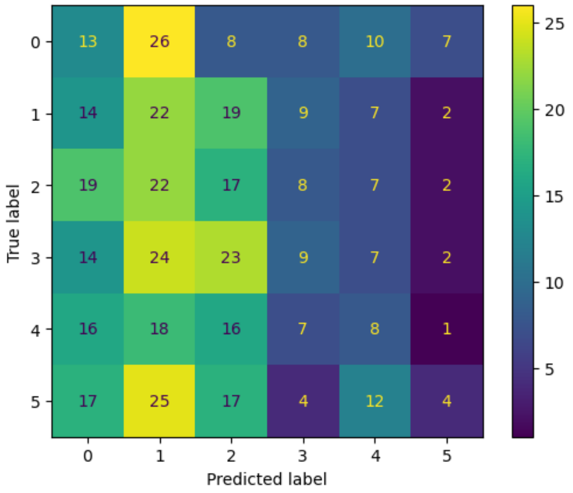
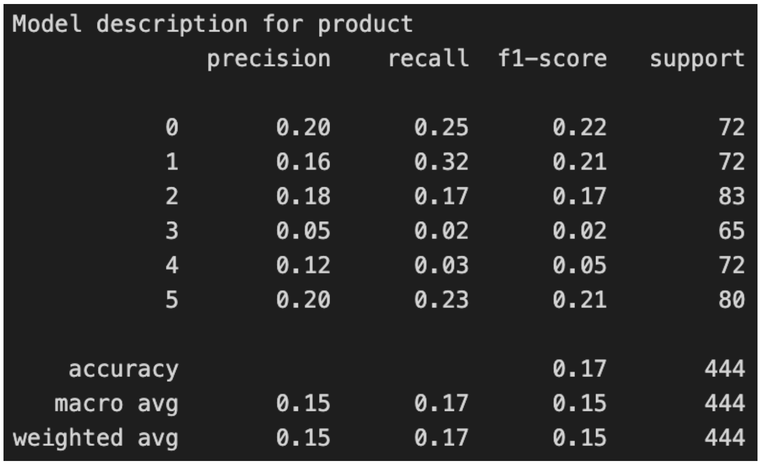
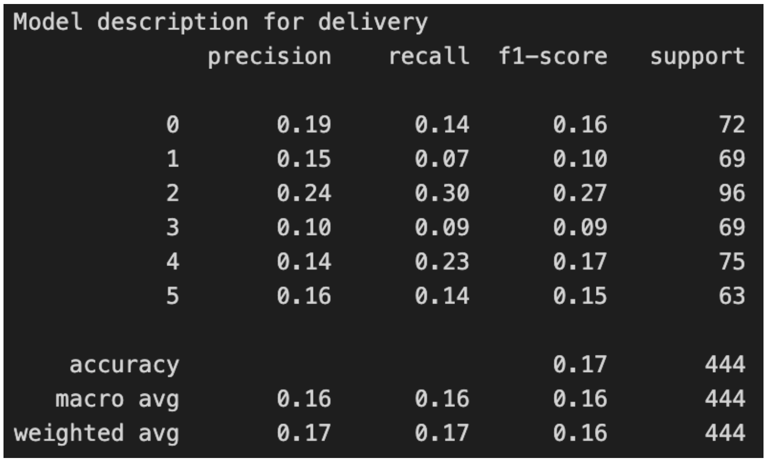


Figure 8. Confusion matrix for RNN (marketplace)

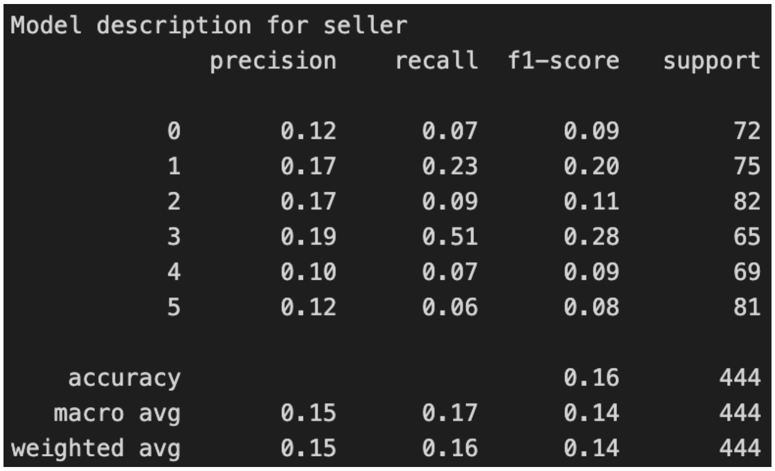
From the confusion matrices it's visible that the models were not capable of generalizing. There are the models descriptions for the best models per aspect



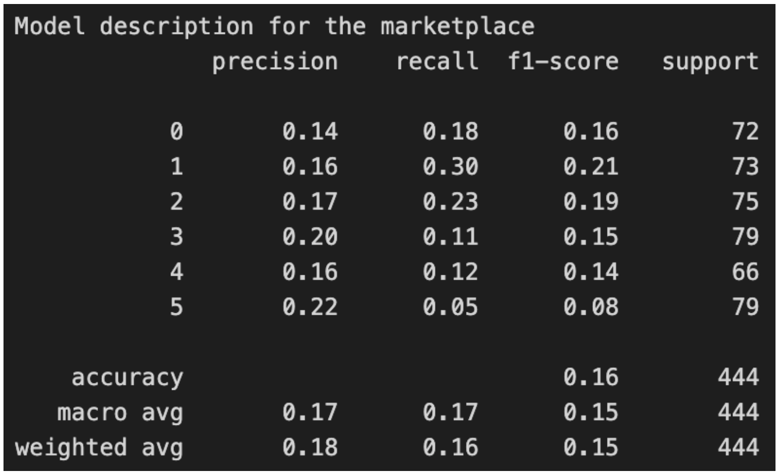
```
Model description for product
              precision    recall  f1-score   support

           0       0.20      0.25      0.22        72
           1       0.16      0.32      0.21        72
           2       0.18      0.17      0.17        83
           3       0.05      0.02      0.02        65
           4       0.12      0.03      0.05        72
           5       0.20      0.23      0.21        80

    accuracy                           0.17       444
   macro avg       0.15      0.17      0.15       444
weighted avg       0.15      0.17      0.15       444
```



```
Model description for delivery
              precision    recall  f1-score   support

           0       0.19      0.14      0.16        72
           1       0.15      0.07      0.10        69
           2       0.24      0.30      0.27        96
           3       0.10      0.09      0.09        69
           4       0.14      0.23      0.17        75
           5       0.16      0.14      0.15        63

    accuracy                           0.17       444
   macro avg       0.16      0.16      0.16       444
weighted avg       0.17      0.17      0.16       444
```



```
Model description for seller
              precision    recall  f1-score   support

           0       0.12      0.07      0.09        72
           1       0.17      0.23      0.20        75
           2       0.17      0.09      0.11        82
           3       0.19      0.51      0.28        65
           4       0.10      0.07      0.09        69
           5       0.12      0.06      0.08        81

    accuracy                           0.16       444
   macro avg       0.15      0.17      0.14       444
weighted avg       0.15      0.16      0.14       444
```



```
Model description for the marketplace
              precision    recall  f1-score   support

           0       0.14      0.18      0.16        72
           1       0.16      0.30      0.21        73
           2       0.17      0.23      0.19        75
           3       0.20      0.11      0.15        79
           4       0.16      0.12      0.14        66
           5       0.22      0.05      0.08        79

    accuracy                           0.16       444
   macro avg       0.17      0.17      0.15       444
weighted avg       0.18      0.16      0.15       444
```

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 pretrained BERT model. As the pretrained 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**	**Marketplace**	**Overall**
31.88%	35.18%	40.18%	33.04%	35.07%

Table 5. Accuracy of the BERT model

The confusion matrices for BERT:

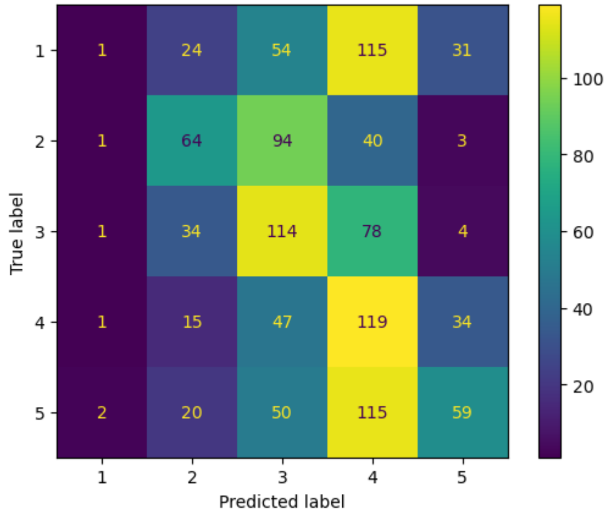


Figure 9. Confusion matrix for BERT (product)

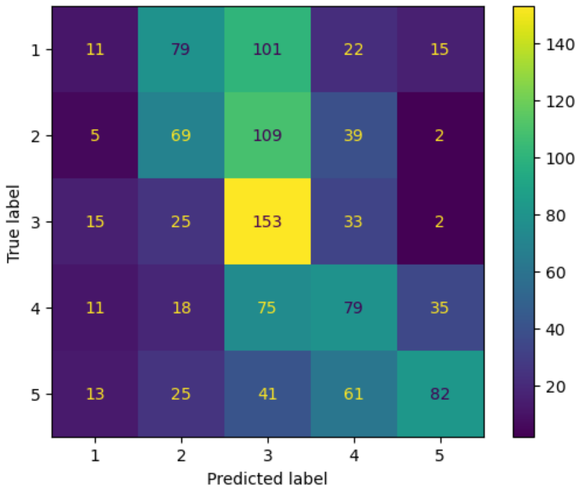


Figure 10. Confusion matrix for BERT (delivery)

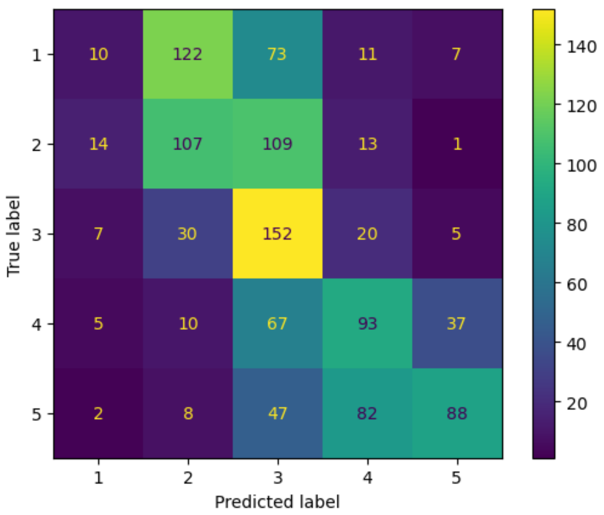


Figure 11. Confusion matrix for BERT (seller)

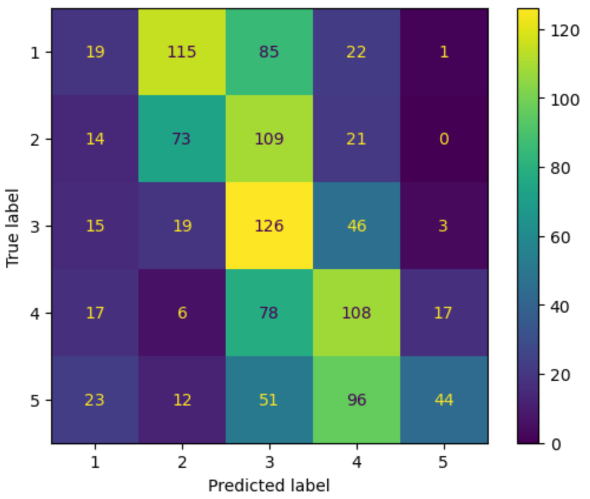


Figure 12. Confusion matrix for BERT (marketplace)

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

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

Product	Delivery	Seller	Marketplace	Overall
34.12%	41.77%	44.70%	43.36%	40.99%

Table 6. Mean accuracy of all models on the test set

that is overall higher by 3%+ than for SVM without aspect splitting.

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.

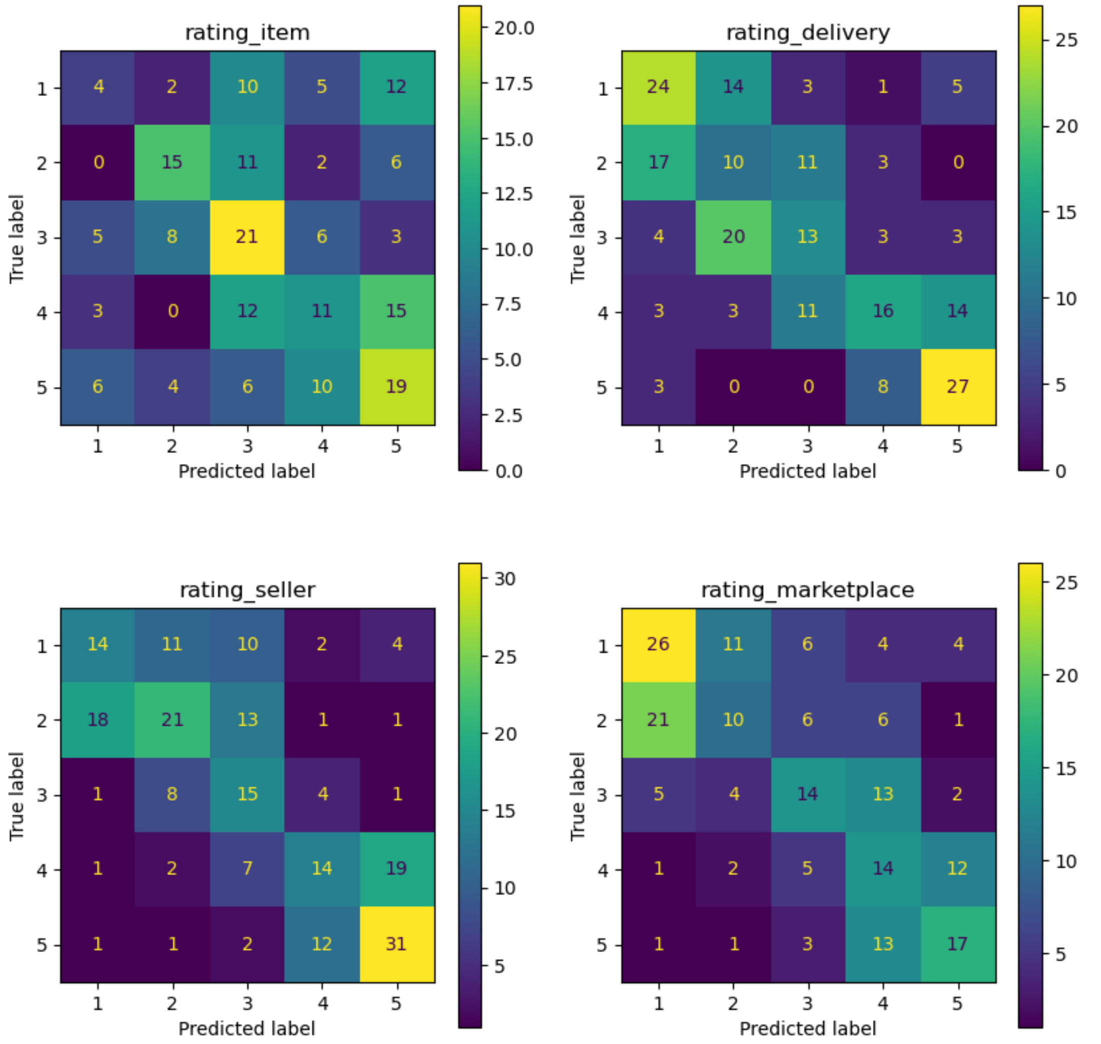
To evaluate and validate models, the noted validation set was used. All trained models were combined into an ensemble with equal weights that allowed to increase the overall accuracy. The achieved accuracy for the ensemble on the validation set

Product	Delivery	Seller	Marketplace	Overall
35.71%	41.67%	44.39%	40.90%	40.47%

Table 7. Accuracy of the ensemble on the validation set

that is overall higher by 1%+ than for SVM without aspect splitting.

The confusion matrices for the ensemble are



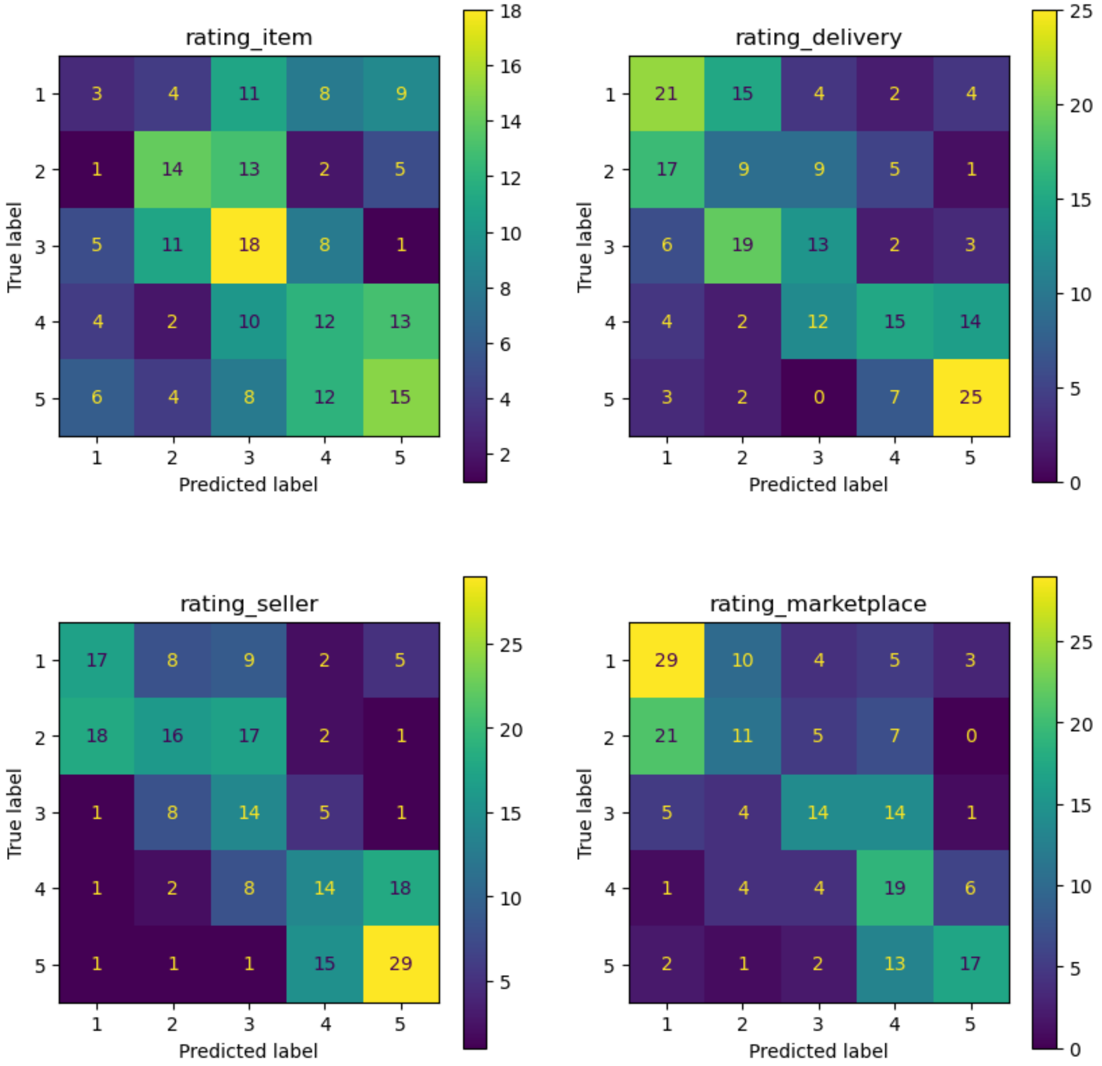
As in one case the accuracy is calculated as a mean for a set of models and in another using the ensemble, it's not possible to evaluate each specific model in detail. To address this issue the best models per aspect were taken. The achieved accuracy for such models on the validation set

Product	Delivery	Seller	Marketplace	Overall
31.16%	38.79%	42.06%	44.56%	39.14%

Table 8. Accuracy of the best models on the validation set

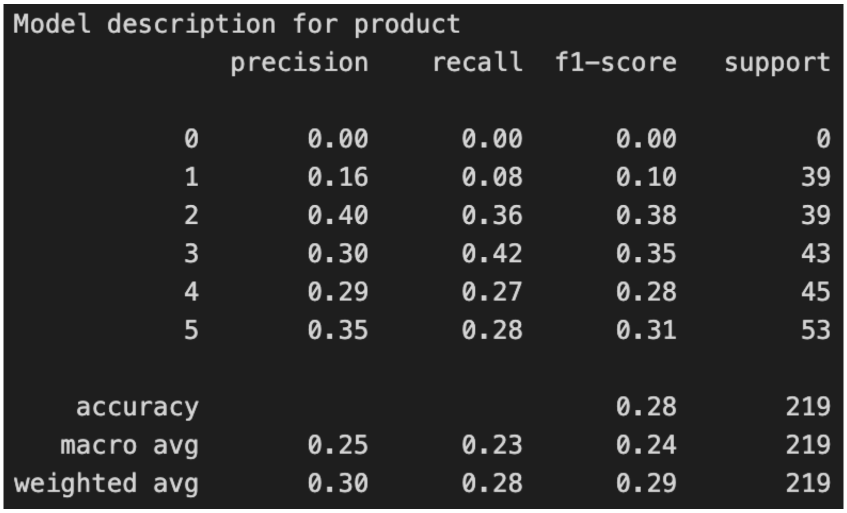
that is overall higher by less than 1% than for SVM without aspect splitting.

The confusion matrices for the best models are

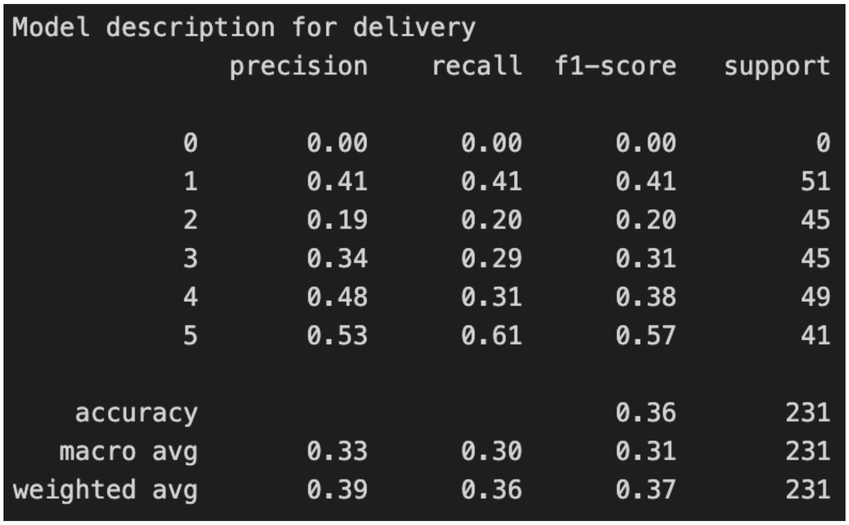


For both the ensemble and the best models from the confusion matrices it’s visible that the models are capable of grasping the general attitude of the reviews, but struggles to recognize a specific rating.

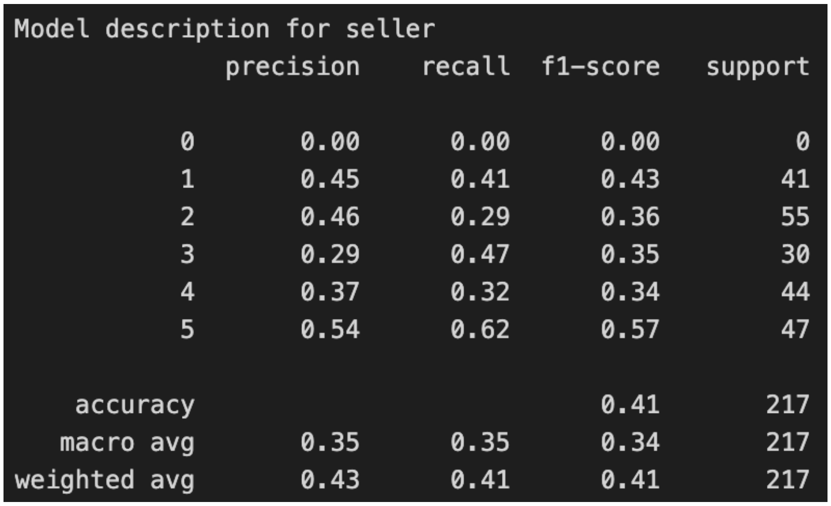
As it’s not possible to give a detailed description for the ensemble, there are the models descriptions for the best models per aspect



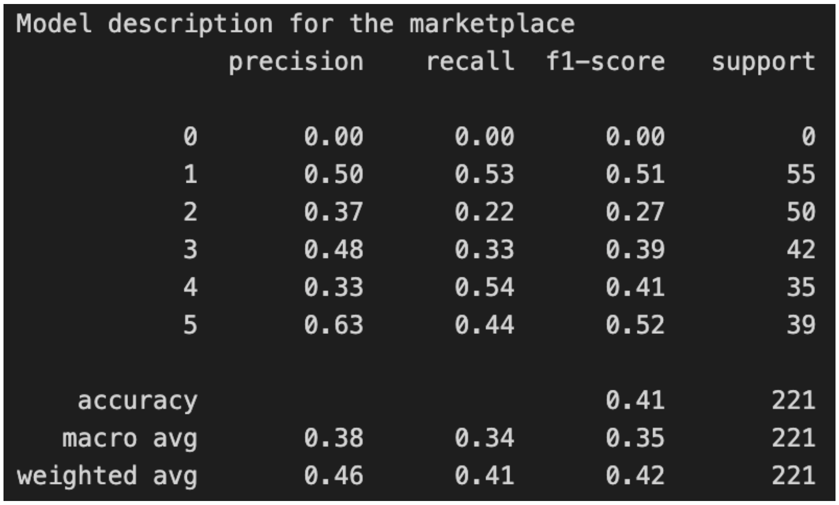
```
Model description for product
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       0.16      0.08      0.10        39
           2       0.40      0.36      0.38        39
           3       0.30      0.42      0.35        43
           4       0.29      0.27      0.28        45
           5       0.35      0.28      0.31        53

    accuracy                           0.28       219
   macro avg       0.25      0.23      0.24       219
weighted avg       0.30      0.28      0.29       219
```



```
Model description for delivery
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       0.41      0.41      0.41        51
           2       0.19      0.20      0.20        45
           3       0.34      0.29      0.31        45
           4       0.48      0.31      0.38        49
           5       0.53      0.61      0.57        41

    accuracy                           0.36       231
   macro avg       0.33      0.30      0.31       231
weighted avg       0.39      0.36      0.37       231
```



```
Model description for seller
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       0.45      0.41      0.43        41
           2       0.46      0.29      0.36        55
           3       0.29      0.47      0.35        30
           4       0.37      0.32      0.34        44
           5       0.54      0.62      0.57        47

    accuracy                           0.41       217
   macro avg       0.35      0.35      0.34       217
weighted avg       0.43      0.41      0.41       217
```



```
Model description for the marketplace
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       0.50      0.53      0.51        55
           2       0.37      0.22      0.27        50
           3       0.48      0.33      0.39        42
           4       0.33      0.54      0.41        35
           5       0.63      0.44      0.52        39

    accuracy                           0.41       221
   macro avg       0.38      0.34      0.35       221
weighted avg       0.46      0.41      0.42       221
```

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 pretrained 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 pretrained 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 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.

Future work

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.