Università della Svizzera italiana

Institute of Computing CI

Data Analytics Group 11

Students: Samuel Corecco, Fabian Gobet mail: corecs@usi.ch, gobetf@usi.ch

Assignment 1

Exploring and describing the data

Our Dataset is made up of 80271 music reviews coming from the website rateyourmusic.com. Each sample has only two fields: a text review and rating score from 0.5 to 5.

Upon initial analysis, we observed a skewed distribution of ratings (Figure 1), predominantly favoring higher ratings (Figures 2a and 2b), reminiscent of a gamma distribution (Figures 11a, 11b).

Following our analysis, we examined the statistical properties of the rating distribution. Table 2 reveals an average rating of 4.25, affirming the existing imbalance. Moreover, with a standard deviation of 0.87, our findings further underscore the prevalence of high ratings within the dataset.

In addition to ratings, we can analyze characteristics of the review text, such as its length in characters. Figure 3 demonstrates a notable variance in length, with a peak in shorter reviews tapering off gradually. These findings are further elucidated in Table 3, offering a statistical overview. The mean length is 907 characters, comparatively modest against the maximum of 32,117. Highlighting the variance within the dataset, the standard deviation stands at 1449.

To deepen our understanding of data processing, we utilized the `langdetect` library to analyze the languages present in the dataset. The findings, illustrated in Figure 4 and summarized in Table 4, indicate the existence of 14 languages, with English comprising the majority at 90.25% (70,551 entries). Given the minimal presence of other languages, managing them would necessitate translation tools, potentially yielding poorly translated English reviews with limited utility. As such, we have opted to exclude them from further analysis.

Table 4 indicates that approximately 0.48% of entries are labeled as an unknown (unk) language, suggesting potential unknown fields within the dataset. While we could exclude these based on English language criteria, it's essential to assess the extent of empty fields overall.

Upon examination, we found that 2.70% of our data (2109 entries) contain unknown fields. Given the significant prevalence of English and the disparity among languages, alongside the asymmetry in rating distribution and weak correlation between rating and sentiment analysis polarity, we've chosen to disregard this small portion of the dataset, ensuring our analysis focuses on a more reliable subset, mitigating potential biases or inaccuracies stemming from unknown or incomplete fields.

Having narrowed our focus to the English language, we analyzed correlations between reviews and their ratings. Using the TextBlob library for sentiment analysis and TextStat for text quality assessment, including Flesch-Kincaid Grade and length, results (Figure 5, Table 5) indicate

no discernible correlations between these parameters and the rating. For a thorough analysis, we employed all functions offered by the TextStat library. Despite this, the results (Figure 6, Table 5) did not unveil any correlations between the rating and review.

Upon scrutiny, it becomes evident that the dataset is notably sparse, consisting solely of reviews and ratings. The absence of details regarding music genres and user profiles poses a significant obstacle to text classification. Without insights into music genres or user preferences, the variability in user responses to identical comments remains unaccounted for. Furthermore, the dataset displays a pronounced bias towards positive reviews. This imbalance not only reduces the representation of lower ratings but also skews the overall dataset towards more favorable evaluations.

A critical observation reveals the absence of any discernible correlation between reviews and their corresponding ratings, which further complicates the classification effort. This challenge is amplified by the dataset's division into ten distinct classes, adding complexity to differentiating subtle distinctions between ratings, such as distinguishing between a 4 and a 4.5 rating.

The dataset's lack of depth, skew towards positive reviews, and the intricacy in distinguishing between closely rated categories collectively escalate the complexity of review classification, presenting a nuanced and challenging task.

Pre-processing the data

Pre-process the data For Bert

Cleaning and embedding data

Given our project's requirements, we opted for utilizing a pre-trained uncased BERT text classification model, considering the insights gleaned from our earlier analysis.

We began by removing entries with unknown or empty fields and filtering for English-based reviews, resulting in a 12.08% reduction in the initial data. Next, we utilized the NLTK library to remove stopwords from the text, except for the word "I," which holds significance in review sentiment. To prepare the reviews for our model, we tokenized the sentences using the pre-implemented BERT tokenizer, truncating them to a maximum of 512 tokens, and employed a data collator with padding to facilitate batching for the model.

As a final step, we linearly mapped the ratings in our dataset, which range from 0.5 to 5.0 in intervals of 0.5, into discrete classes. Given that there are 10 different ratings, we mapped each rating to a class index ranging from 0 to 9.

Data split and criterion for BERT

We utilized Scikit-learn's train-test stratified split method to achieve an 80%-10%-10% split for the train, validation, and test sets, respectively. Considering the dataset's imbalance, one viable approach to mitigate it is through the use of weighted criteria. Therefore, we initially opted to use the dataset as it stands, incorporating said pre-processing and custom weighted loss function.

Pre-process the data For XGBoost

Data augmenting

Due to the unbalanced nature of the dataset, we have decided to use the `nlpaug.augmenter.word` library for data augmentation. Balancing the dataset would introduce distortions, so instead we have increased the number of reviews in all classes to a minimum of 2000.

To address our unbalanced data, we will combine augmentation with class weights to limit the problem without creating distortion.

Cleaning and embedding data

We have removed any reviews that are empty or not in English during the text cleaning process.

Different TF-IDF (Term Frequency-Inverse Document Frequency) representations were adopted for data embedding.

- **TF-IDF at word level** This strategy attempts to capture the importance of each word in the data compared to its frequency in specific reviews. This step is used to identify significant terms that can directly influence the rating assigned to a review.
- **TF-IDF at n-gram level** The second strategy is similar to the previous one, but instead of looking for single words, it tries to capture a context. This step therefore helps to look for combinations of words that, when used together, give a different meaning that single words cannot express, e.g. "I hate this music" and "I hate those who rate this music badly".
- **TF-IDF at character level** Similar to the previous strategy, but considering characters, it aims to give more details that try to capture information such as word roots, suffixes, prefixes, ... This is meant in the context of the review, as the text may contain typos and slang.

Data split for XGBoost

The data split was done with 80% training, 10% validation and 10% testing

Building a model for rating prediction

BERT

Weighted criterion

To address the issue of class imbalance, we experimented with training a model using a weighted criterion tailored to the imbalance present in the classes. Initially, we utilized Scikit-learn's `compute_class_weight` method to calculate a weight vector that ensures a balanced proportion of ratings within the training dataset.

With the weight vector established, we then implemented a custom class inheriting methods from the Hugging Face Trainer class. Within this custom class, our focus was on implementing the `compute_loss` method to incorporate the weighted criterion based on the multi-label cross-entropy function, thus theoretically enabling the model to learn effectively while minimizing bias and enhancing predictive accuracy.

Regarding the training arguments, we found that the most suitable parameters were as follows:

- Initial learning rate: 2×10^{-4}.
- Learning rate scheduler: Linear with a warmup of 500 steps.
- Weight decay: 1×10^{-3}.
- Batch size: 16.
- Gradient accumulation: 8 steps.
- FP16 flag set to true for float-16 quantization to reduce training time without significant performance loss.
- Number of epochs: 5, with the option to load the best model at the end of training to mitigate potential overfitting.

Through experimentation and optimization, we carefully selected parameters to balance training efficiency and model performance.

Analysis of Figure 7 indicates that overfitting begins after the second epoch, suggesting a potential need for early stopping. However, Figure 8 reveals that the highest evaluation accuracy is attained at the end of epoch 3.

Despite the onset of overfitting after epoch 2, the losses in epoch 3 closely resemble those of epoch 2. Thus, epoch 3 still demonstrates promising performance metrics.

Considering all factors, the best model achieved using this methodology is the one obtained at the end of epoch 3, with an evaluation accuracy of 42

Standard criterion

After finding limited promise in the weighted criterion BERT model, we explored the performance of a standard unweighted multi-label cross-entropy loss criterion. The only change made was in the criterion used, while all other aspects of the model configuration remained the same.

During training with the unweighted multi-label cross-entropy loss criterion, overfitting occurred similarly in epoch 2, while epoch 3 exhibited the best performance. Compared to the previous version utilizing the weighted criterion, we achieved an accuracy of 46% with this model.

However, it's important to note that we cannot immediately conclude that this model is superior. Further analysis, including examination of the confusion matrix on the test sets for both models, is necessary. This comprehensive evaluation will provide deeper insights into the performance and effectiveness of each model, facilitating a more informed comparison and selection of the optimal approach.

XGBoost

As previously stated, we determined the weights for the different classes using the `class_weight` library. To avoid risks of overfitting we decided to optimize some specific parameters of XGBoost (see Table 7)

Training XGboost

For training we used "mlogloss", a logarithm of the multiclass loss. Furthermore we also use, as mentioned previously, the `compute_sample_weight('balanced', y_train)`, to manage the imbalance of the classes

Model evaluation

BERT - standard and weighted criterion

The BERT-based models' training results are summarized in Table 6. Despite utilizing a weighted criterion to address the dataset's imbalance, further investigation into the implications of these accuracies and their differences is imperative. This examination is conducted in light of the respective confusion matrices depicted in Figures 11 and 12.

Upon analysis, it's observed that the increase in accuracy of the standard model is attributed to a rise in true positives for predictions of data belonging to the dense distribution parts of the original data, particularly higher ranges of ratings. However, in regions of the dataset characterized by lower ratings, numerous misclassifications surpass the imbalance proportion.

The weighted version of the model mitigates this problem in favor of overall accuracy. However, the issue of imbalanced data remains prominent, particularly affecting predictions for lower range ratings.

XGBoost

The analysis of our XGBoost model was carried out after training. The results are presented in Table 8, and the confusion matrix is shown in Figure 14. The accuracy obtained for our test set was 38%, indicating that the XGBoost model did not perform as well as BERT in terms of overall efficiency.

The confusion matrix confirmed the hypotheses made during the data analysis: the distinction between classes is unclear, and the low-rated classes have insufficient data. A diagonal pattern is visible for classes rated 2.5 and above, while lower classes have almost no predictions, similar to the outcome obtained with BERT, albeit slightly less precise.

However, the most important aspect is not the overall performance of the model, but rather its reliability in making predictions. We examined the degree of confidence in XGBoost's predictions and found that it is a reliable and significant indicator (refer to Figure 13). When we segmented the predictions based on confidence, we discovered that the model achieved an accuracy of 76% on a sample that was reduced to 10% of the data under high confidence conditions (confidence greater than 0.7). The training results, as shown in Figure 15, present a matrix with a more defined diagonal, which supports our previous observations. Although the results are promising, it is important to note that the model currently struggles to effectively handle lower ratings. This limitation is due to the aforementioned shortcomings.

Discussion

The prediction models in this project were built under the hypothesis that the available data would be sufficient for accurate predictions. However, upon analysis, which included employing various tools such as sentiment inference, it became evident that the correlation between the review text and the rating was very weak. This weakness was reflected in the highly skewed distribution of ratings.

Further investigation, aided by sentiment analysis, revealed instances where reviews with negative sentiment polarity were associated with high ratings. This suggests that individuals may prioritize the content of their reviews over assigning an appropriate rating. This observation contradicted our initial hypothesis.

Despite employing advanced AI technologies and training various models, the highest accuracy achieved was nearly 50% on the testing dataset, or 75% on only 10% of the dataset with a confidence above 0.7. This outcome indicates that the dataset lacks features necessary for accurate prediction.

In conclusion, the current findings refute our initial hypothesis, highlighting the insufficiency of the dataset for generating accurate predictions.

Appendix

.1. Data analysis and pre proccessing

Review	Rating
i think i actually under-rate ok computer if anything, the back half is just as strong as the first	5.0
i get why radiohead rub a lot of people the wrong way. thom yorke's voice is an acquired taste.	5.0
i would like to think i am good about not letting hype affect my judgment of music.	4.5
there are radiohead devotees like there were once grateful dead followers.	4.0
i wrote a shining excellent review for this album before i deleted my rym to get a fresh start.	5.0

Table 1: Example of our dataset.

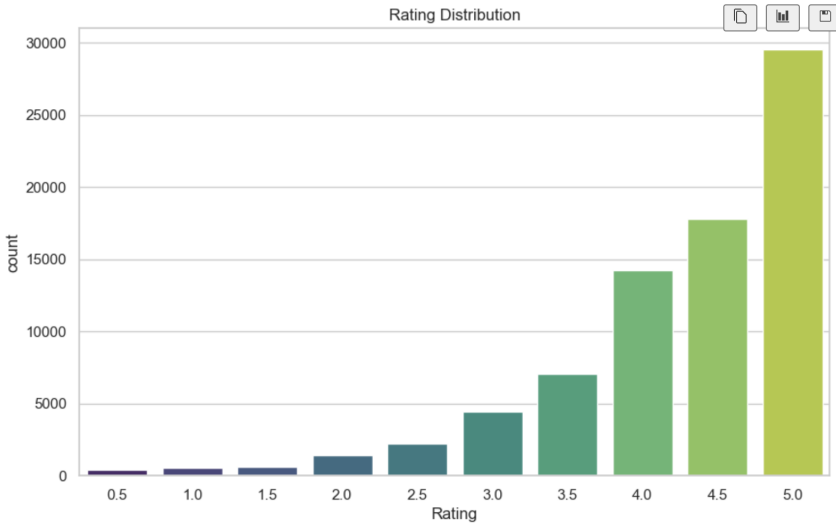
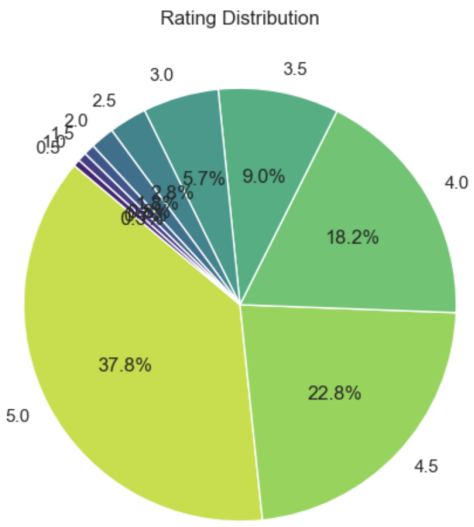


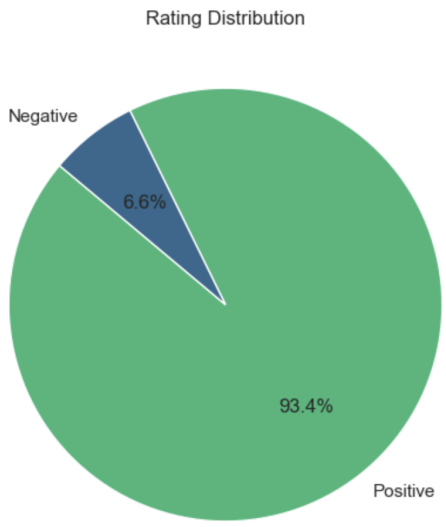
Figure 1: distribution of reviews with respect to rating

Statistic	Value
Mean	4.253211
Standard Deviation	0.870954
Minimum	0.500000
25% Percentile	4.000000
Median (50%)	4.500000
75% Percentile	5.000000
Maximum	5.000000

Table 2: Descriptive Statistics of Evaluations



(a) distribution by percentage of ratings



(b) distribution by percentage of ratings

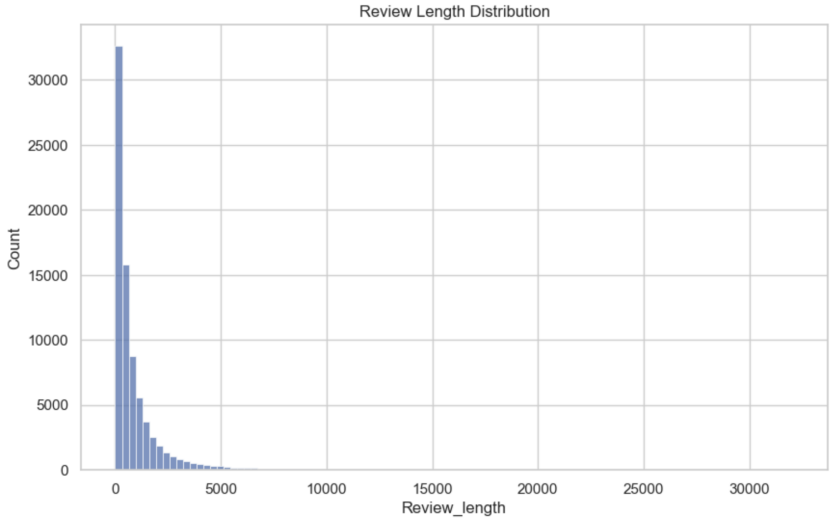


Figure 3: distribution of reviews with respect to length

Statistic	**Value**
Mean	907.8471
Median	428.0
Standard Deviation	1449.8694
Maximum Length	32117
Minimum Length	1

Table 3: Statistic information about length of reviews

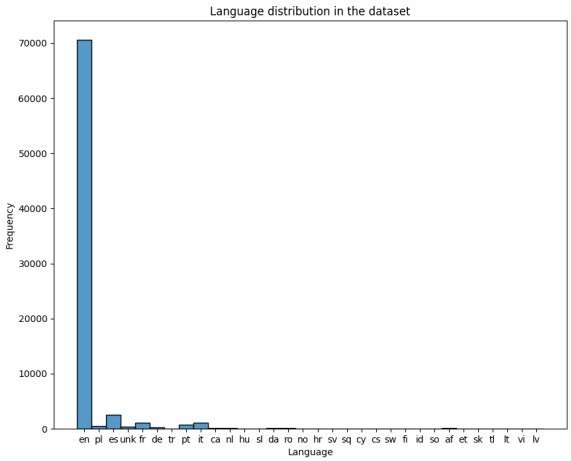


Figure 4: Distribution of languages in the dataset

Language	**Percentage**
en	90.26
es	3.26
fr	1.39
it	1.36
pt	0.91
pl	0.69
unk	0.48
de	0.33

Language	**Percentage**
da	0.18
nl	0.17
ro	0.13
af	0.09
ca	0.08
no	0.07
hu	0.07

Table 4: Language frequencies

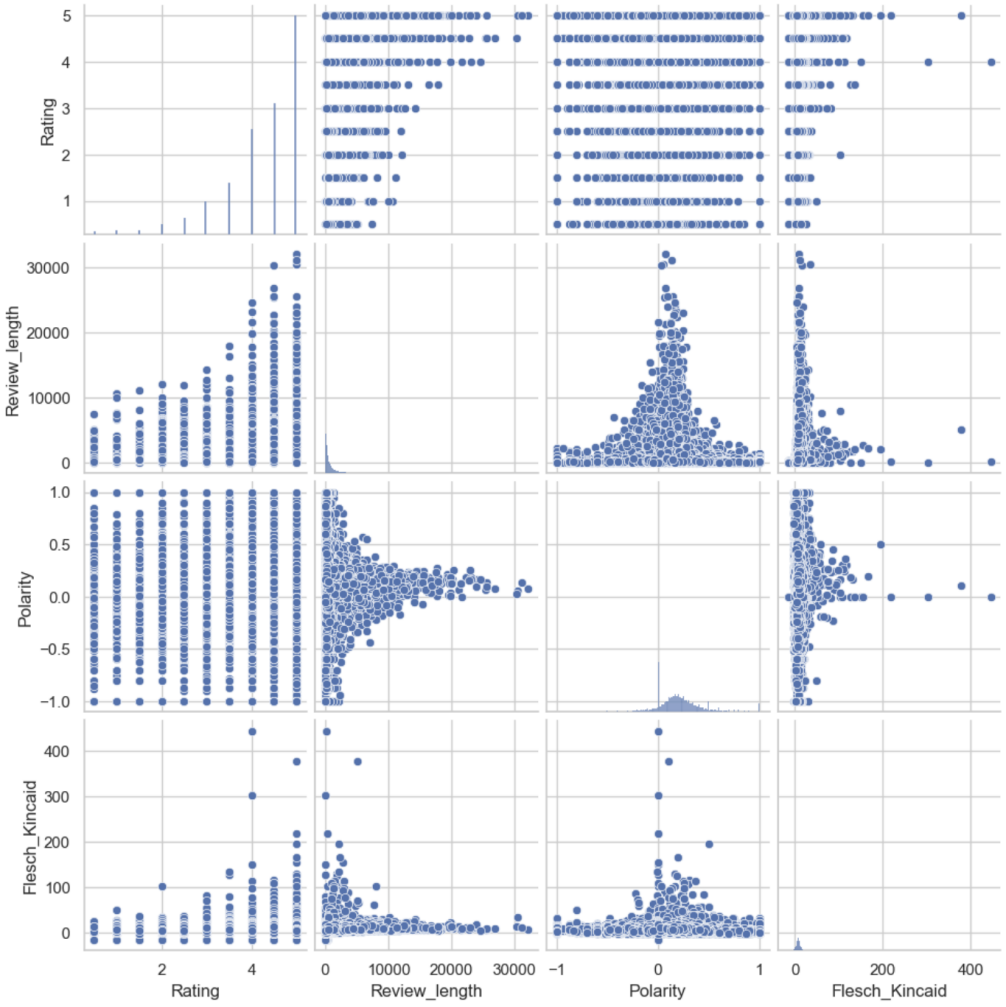


Figure 5: Correlation rating and review 1

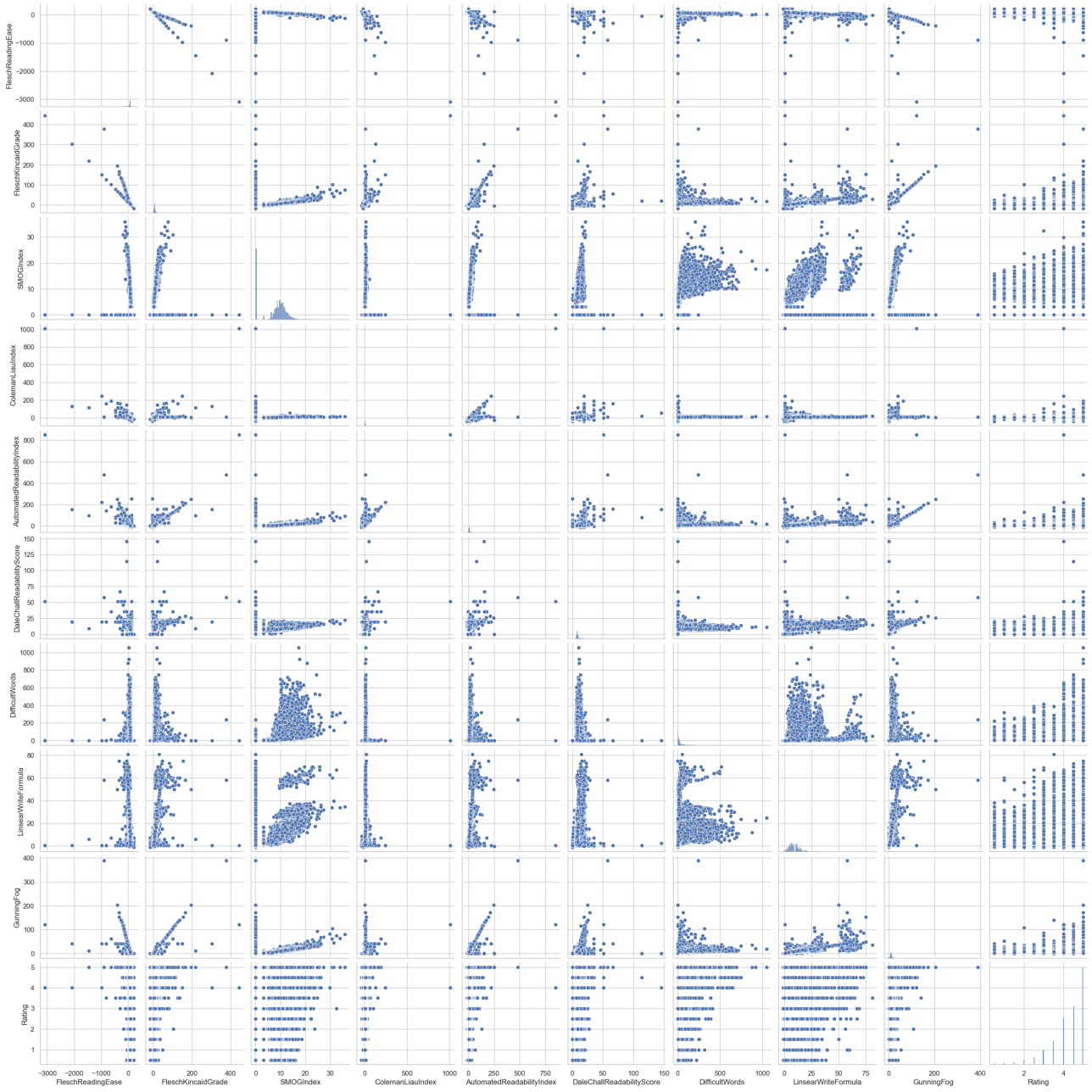


Figure 6: Correlation rating and review 2

Metrics	**Correlation with Rating**
length of the review	0.03698
Sentimental analys	0.20272
Flesch Reading Ease	-0.016709
Flesch Kincaid Grade	-0.001011
SMOG Index	-0.023629
Coleman Liau Index	0.012718
Automated Readability Index	0.005575
Dale Chall Readability Score	0.021587
Difficult Words	0.044004
Linsear Write Formula	-0.033096
Gunning Fog	-0.011556

Table 5: Correlation between Metrics and Ratings

.2. Bert standard and weighted

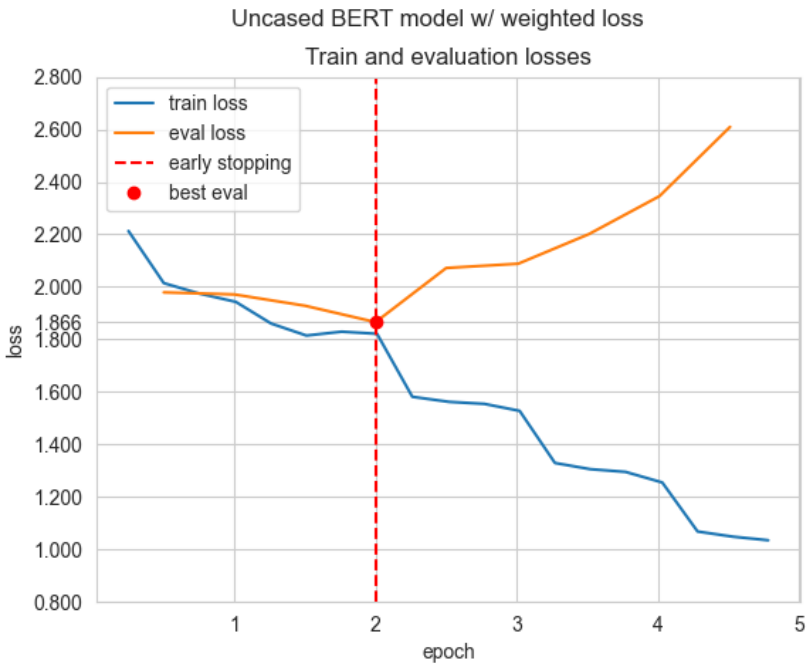


Figure 7: Weighted BERT losses

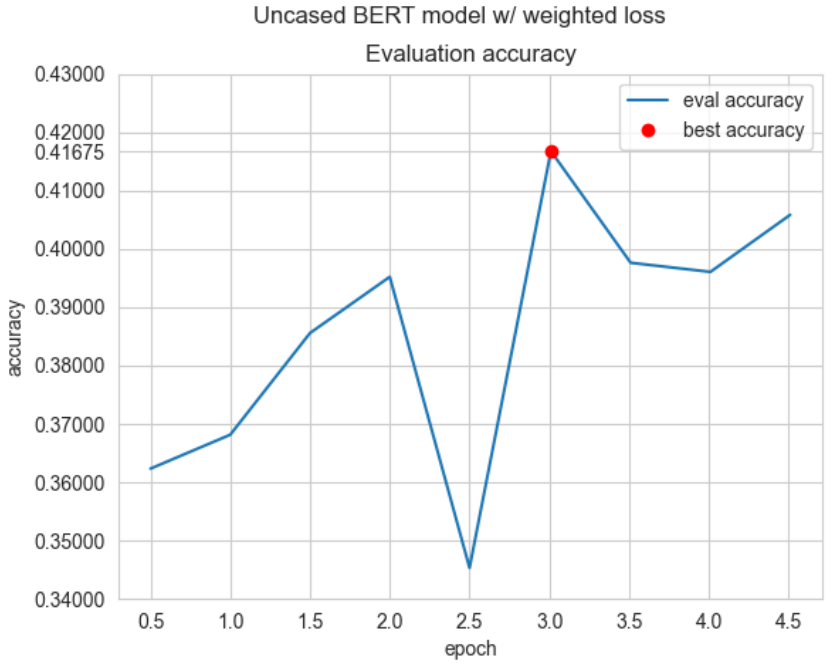


Figure 8: Weighted BERT evaluation accuracies

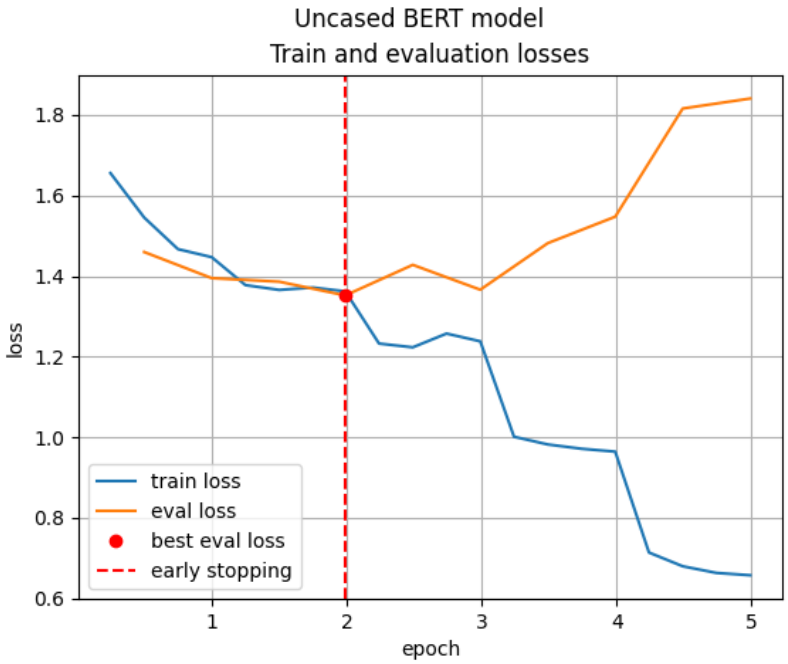


Figure 9: BERT losses

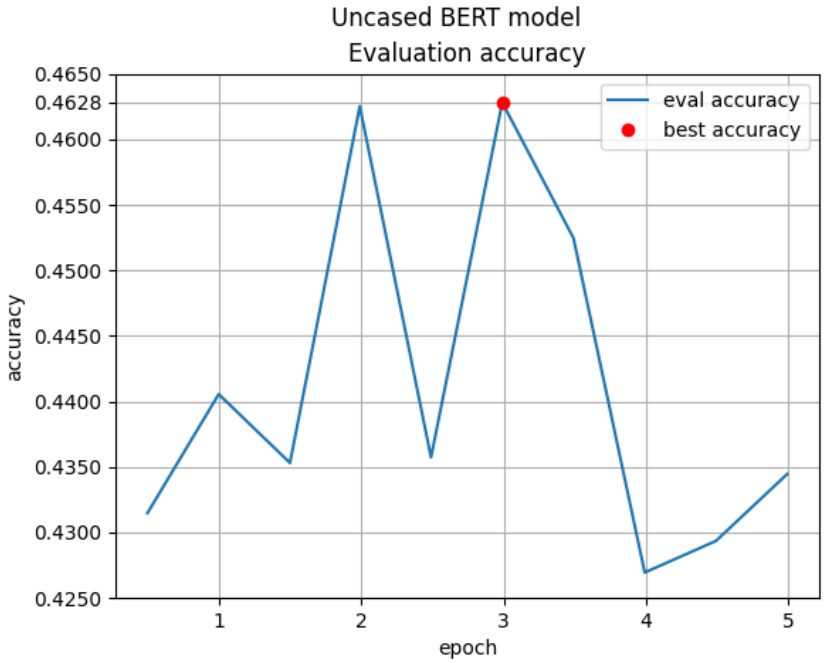


Figure 10: BERT evaluation accuracies

Model	**Accuracy**
Standard BERT	41%
Weighted Bert	46%

Table 6: Test set accuracies for BERT based models

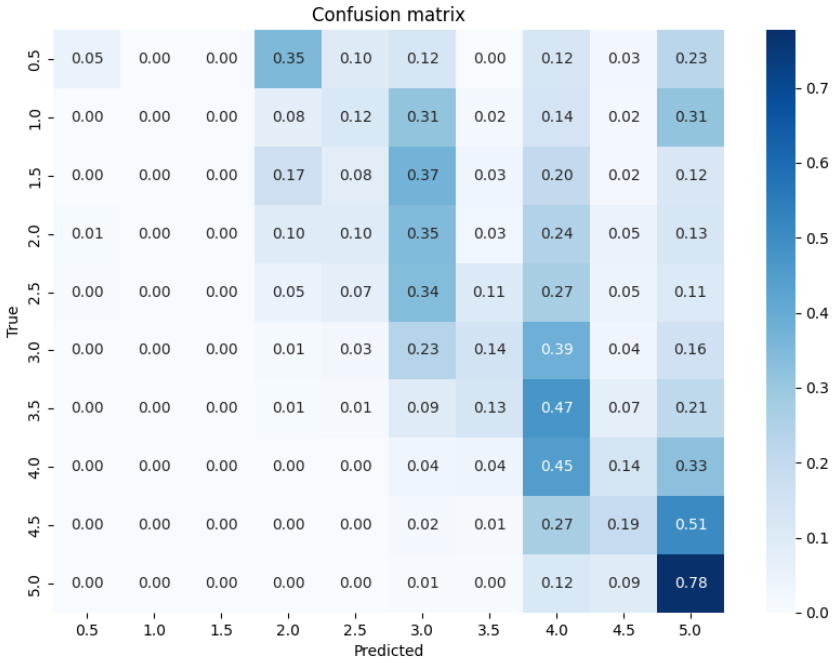


Figure 11: BERT confusion matrix

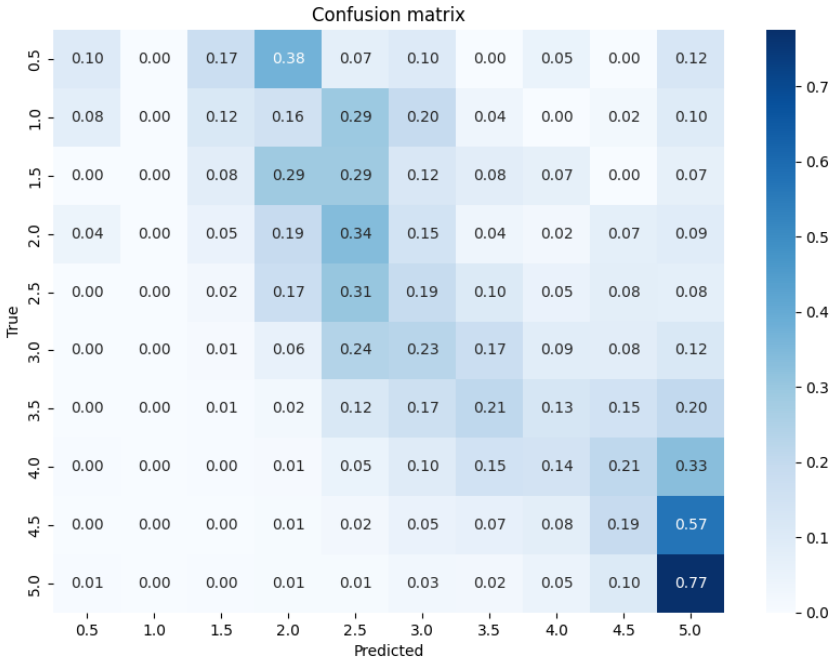


Figure 12: BERT weighted confusion matrix

Variable	Value
max_depth	8
min_child_weight	1
gamma	0.3
subsample	0.8
colsample_bytree	0.8
reg_alpha	0.01
reg_lambda	1
early_stopping_rounds	10

Table 7: Parameters for XGboost

.3. XGBoost

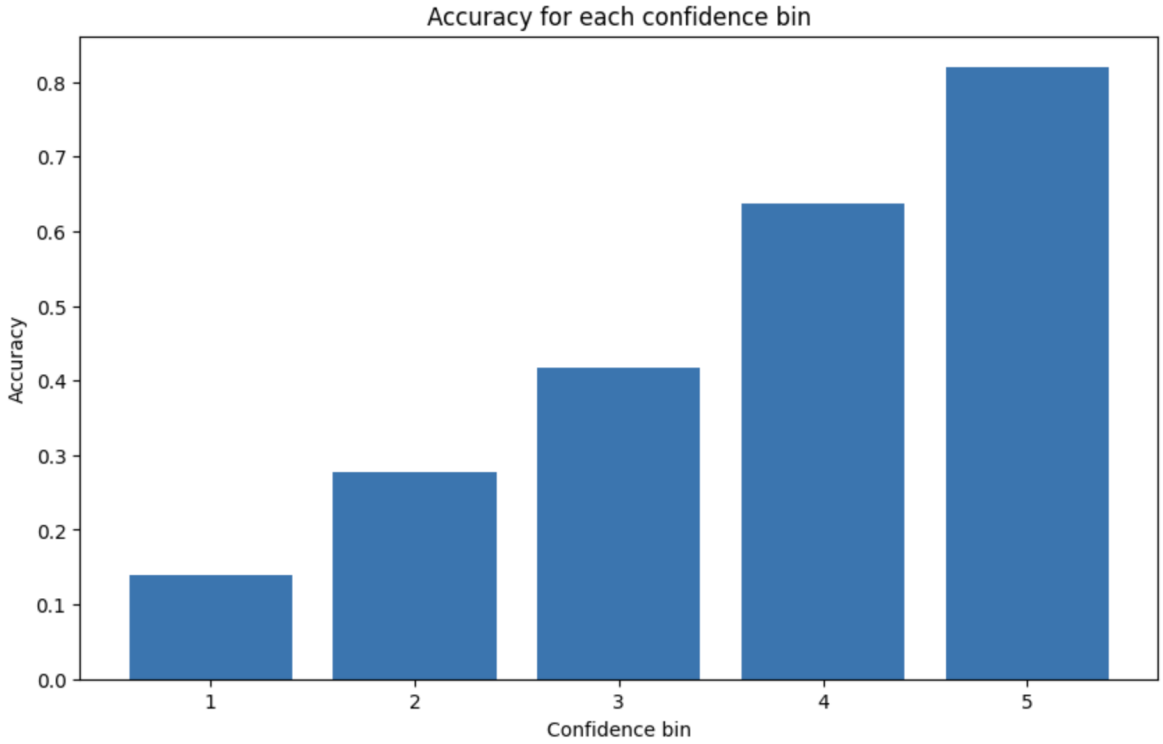


Figure 13: Confidence vs Prediction

Model	Accuracy
XGBoost	38%
XGBoost with confidence > 0.7	76%

Table 8: Test set accuracies for XGBoost models

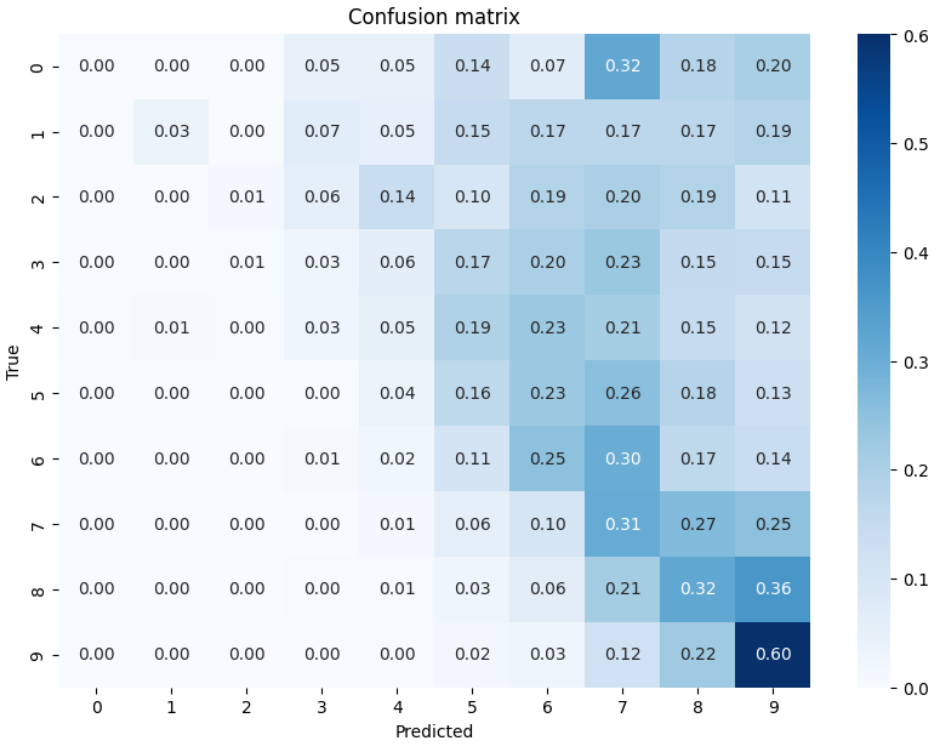


Figure 14: XGBoost confusion matrix

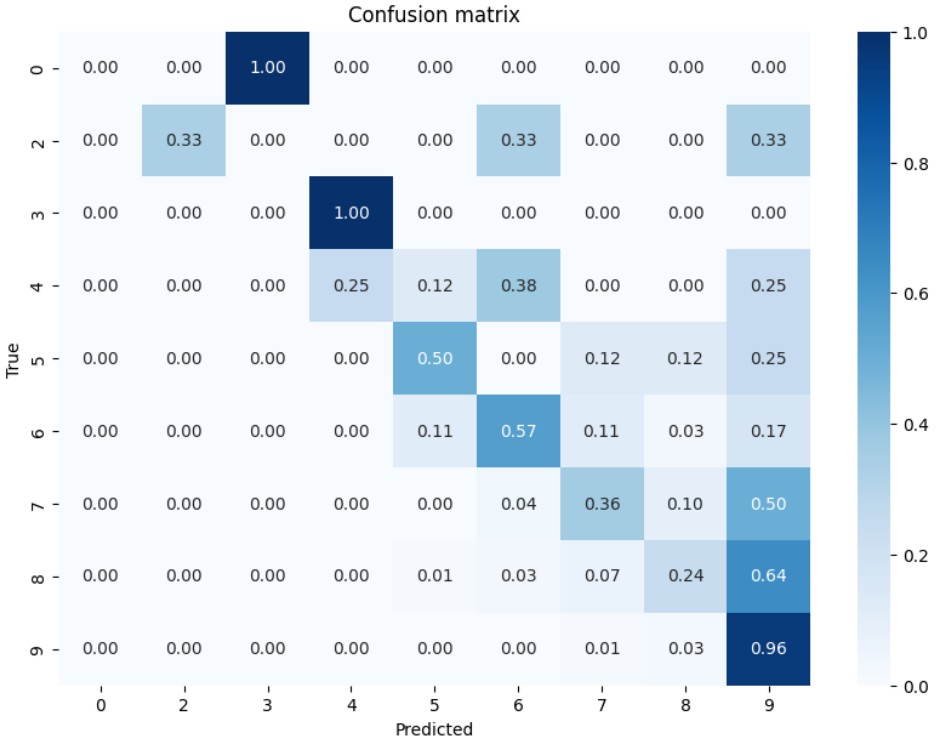


Figure 15: XGBoost confusion matrix with confidence > 0.7