F20AA : APPLIED TEXT ANALYTICS: COURSEWORK 2

UG 3 Dubai

About

This project applied advanced text analytics and machine learning techniques to predict amazon review ratings and extract meaningful topics from the reviews.

Ahmad Meda, Ratish Palanisamy, Mohamed Zayan & Mohamed Kamel

Student Declaration of Authorship

Course code and name:	F21AA - Applied Text Analytics
Type of assessment:	**Group**
Coursework Title:	Cw2 (Amazon Review) -UG
Student Name:	Ahmad Meda
Student ID Number:	H00414901

Declaration of authorship. By signing this form:

- **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.

- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.

- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(type your name):* *Ahmad Meda*

Date: *07/04/2025*

Copy this page and insert it into your coursework file in front of your title page.
For group assessment each group member must sign a separate form and all forms must be included with the group submission.

Your work will not be marked if a signed copy of this form is not included with your submission.

Student Declaration of Authorship

HERIOT WATT UNIVERSITY
UK | DUBAI | MALAYSIA

Course code and name:	F21AA - Applied Text Analytics
Type of assessment:	**Group**
Coursework Title:	Cw2 (Amazon Review) -UG
Student Name:	Mohamed Kamel
Student ID Number:	H00401385

Declaration of authorship. By signing this form:

- I **declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.

- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.

- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(type your name)*: *Mohamed Kamel*

Date: *07/04/2025*

Copy this page and insert it into your coursework file in front of your title page.
For group assessment each group member must sign a separate form and all forms must be included with the group submission.

Your work will not be marked if a signed copy of this form is not included with your submission.

Student Declaration of Authorship

HERIOT WATT UNIVERSITY
UK | DUBAI | MALAYSIA

Course code and name:	F21AA - Applied Text Analytics
Type of assessment:	**Group**
Coursework Title:	Cw2 (Amazon Review) -UG
Student Name:	Ratish Palanisamy
Student ID Number:	H00388361

Declaration of authorship. By signing this form:

- **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.
- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.
- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(type your name):* *Ratish Palanisamy*

Date: *07/04/2025*

Copy this page and insert it into your coursework file in front of your title page.
For group assessment each group member must sign a separate form and all forms must be included with the group submission.

Your work will not be marked if a signed copy of this form is not included with your submission.

Student Declaration of Authorship

HERIOT WATT UNIVERSITY
UK | DUBAI | MALAYSIA

Course code and name:	F21AA - Applied Text Analytics
Type of assessment:	Group
Coursework Title:	Cw2 (Amazon Review) -UG
Student Name:	Mohammed Zayan Shameer
Student ID Number:	H00414337

Declaration of authorship. By signing this form:

- **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.
- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.
- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(type your name):* *Mohammed Zayan Shameer*

Date: *07/04/2025*

Copy this page and insert it into your coursework file in front of your title page.
For group assessment each group member must sign a separate form and all forms must be included with the group submission.

Your work will not be marked if a signed copy of this form is not included with your submission.

Data Exploration and Visualisation

The exploratory analysis of the Amazon Reviews dataset showed that it contains 309,131 reviews in the training set and 119,662 reviews in the test set for food products. Each review in the training set has two columns, a text description (Text) and a rating (Score) from 1 to 5, while the test set includes an Id and Text for prediction in a Kaggle competition. The analysis of the dataset will help understand the dataset's structure, identify imbalances, analyse review lengths, and explore common word usage through word clouds. These insights will be critical for building an effective text classification system to predict ratings based on reviews.

Discussions

The training set contains 309,131 reviews, with ratings ranging from 1 to 5, and the test set contains 119,662 reviews for prediction.

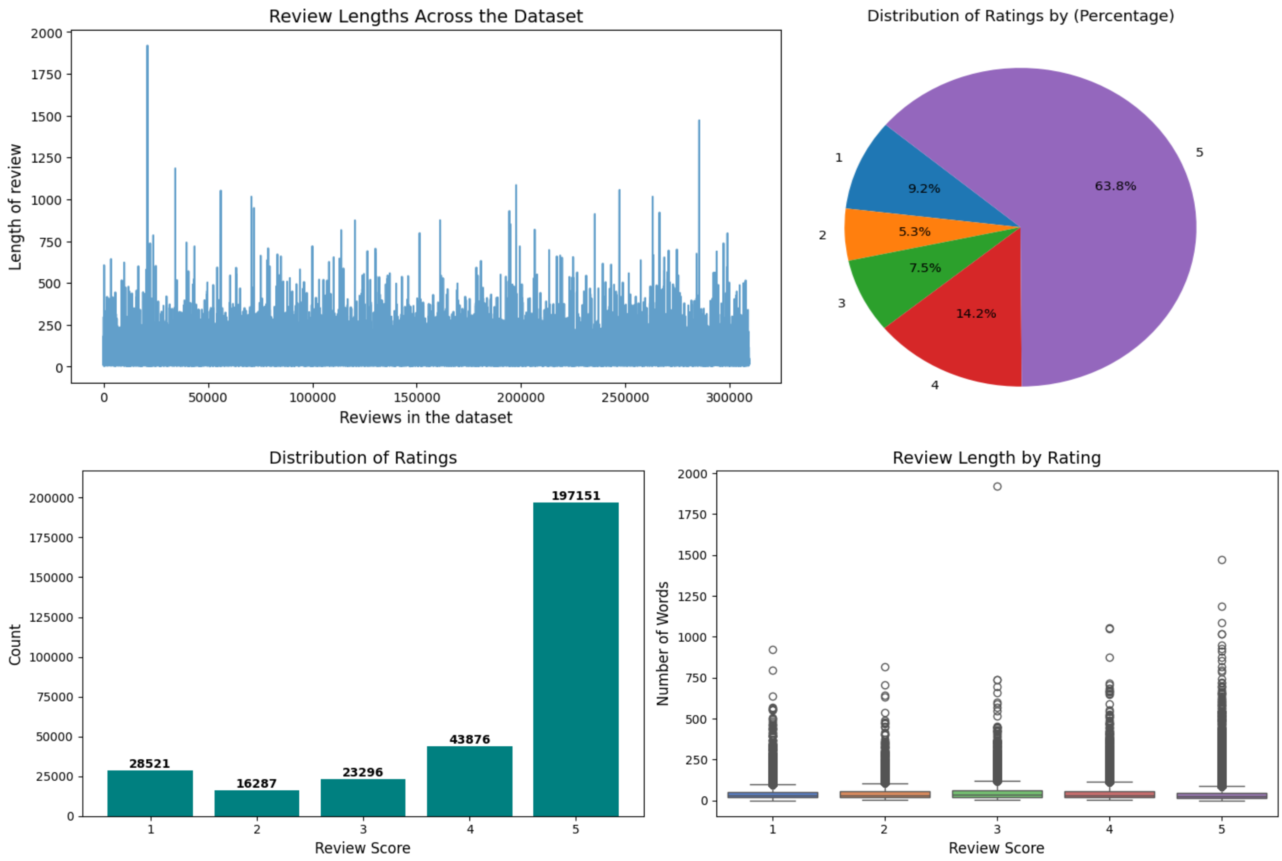
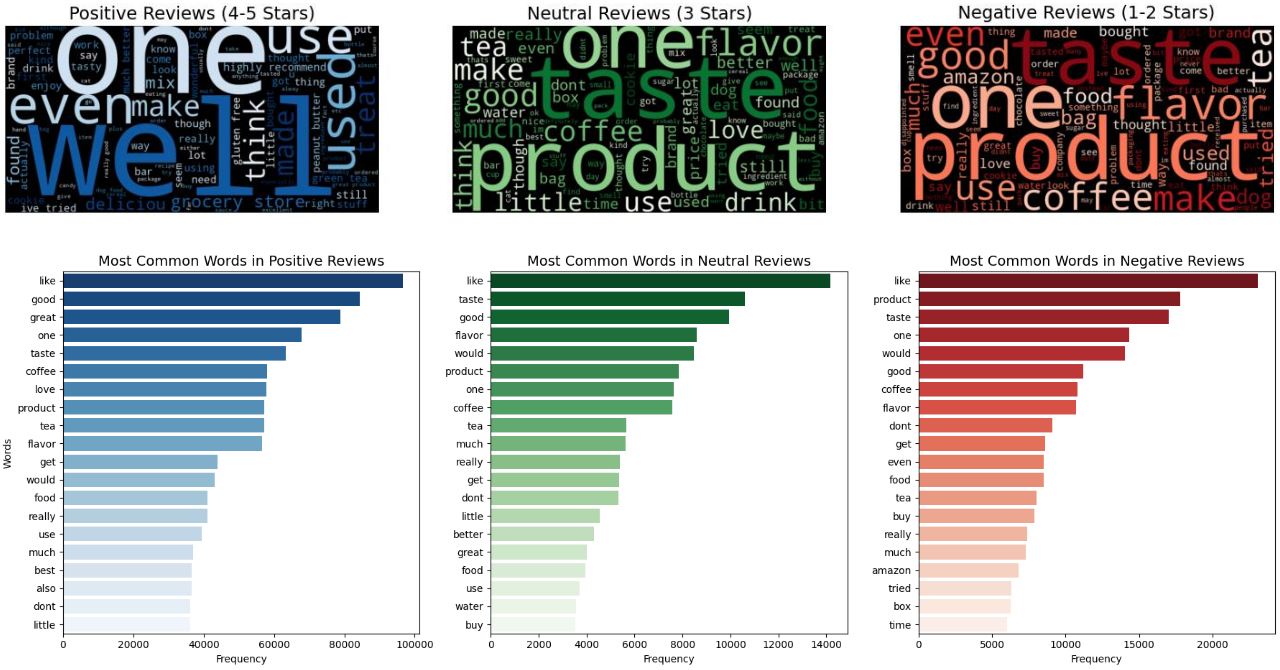


Figure 1: Distribution of Ratings, Distribution of Ratings (Percentage), Review Length by Rating and Review Length across the Dataset

Figure 1 shows the Distribution of Ratings, Distribution of Ratings (Percentage), Review Length by Rating and Review Length across the Dataset. The Distribution of Ratings bar chart and pie chart show a significant skew with 63.8% of reviews (197,151) rated 5 stars, while only 9.2% (28,521) are rated 1 star, and 2 star and 3 stars have even lower distribution. This imbalance suggests that models may overpredict higher ratings, which could potentially lead to poor performance on lower rated reviews.

Review Lengths Across the Dataset histogram indicates that most reviews are short, with a median length but some exceed 1,500 words. Review Length by Rating further shows that 5-star reviews have more variability and goes far beyond when compared to lower ratings, which are generally shorter.



Most Common Words in Positive, Neutral, and Negative Reviews (bar charts) and Word Clouds (all reviews, positive, neutral, negative) provide insights into word usage and are shown in the above figure. The word cloud for all reviews highlights frequent terms like “one”, “well”, “think”, and “use”, which are very generic. The bar charts and word clouds show sentiment specific patterns, positive reviews have words like “great” and “love”, neutral reviews include “flavor” and “product”, and negative reviews feature “bad” and “problem” which indicates dissatisfaction.

Overall, the data exploratory analysis provides several insights to guide the next steps. The heavy skew toward 5-star ratings indicates a need to address class imbalance in the model training phase, possibly through oversampling or class weighting.

Text Processing and Normalisation

Text processing and normalization are crucial in preparing the dataset for machine learning models. In this study, we employed several preprocessing techniques to clean and normalize the text data, including lowercasing, tokenization, removal of stopwords, punctuation, and lemmatization.

We checked the dataset for duplicates to make sure that the model is trained properly and doesn’t cause overfitting the training dataset. After preprocessing we found out that the there were 884 duplicate instances.

We divided our text processing into two parts. One for Machine Learning Models and an other for Transformers.

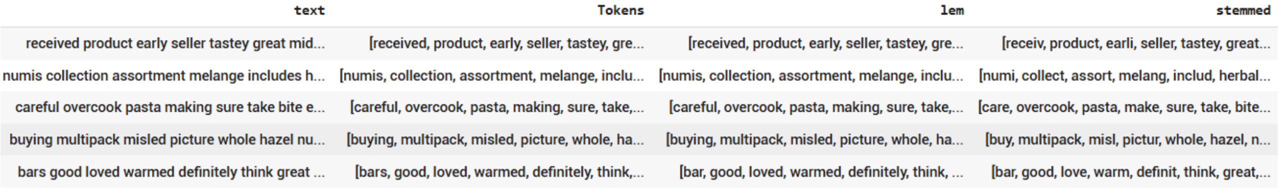
When working with transformer models such as BERT or GPT, extensive cleaning of the text is often not required, unlike Machine Learning Models. These models are pre-trained on vast amounts of raw text, which means they are inherently robust to variations in punctuation, casing, and minor formatting inconsistencies. In other words, they have learned to understand context and semantics from text that retains much of its original structure.

Preprocessing for Machine Learning Models

Techniques Used:

- Lowercasing: All text data was converted to lowercase to ensure consistency, as models should treat "Apple" and "apple" as the same word.
- Tokenization: Text was split into individual words or tokens. This step helps models understand the structure of the text.
- Stopword Removal: Common, non-informative words (such as "the", "and", "is") were removed, which helped reduce the dimensionality of the feature space without losing significant meaning.
- Lemmatization: Words were reduced to their root form (e.g., "running" to "run"), ensuring that different forms of a word were treated as the same.
- Vectorization: We experimented with CountVectorizer and TF-IDF, which represent the frequency of words in the text, capturing critical information about the data.

The image below shows the text cleaned tokenized lemmatized and stemmed for training. This is necessary for getting accuracy when training ML Models and predictions. Our results discuss this topic in detail below.

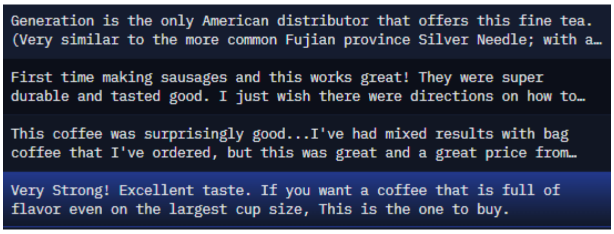


text	Tokens	lem	stemmed
received product early seller tastey great mid...	[received, product, early, seller, tastey, gre...	[received, product, early, seller, tastey, gre...	[receiv, product, earli, seller, tastey, great...
numis collection assortment melange includes h...	[numis, collection, assortment, melange, inclu...	[numis, collection, assortment, melange, inclu...	[numi, collect, assort, melang, includ, herbal...
careful overcook pasta making sure take bite e...	[careful, overcook, pasta, making, sure, take,...	[careful, overcook, pasta, making, sure, take,...	[care, overcook, pasta, make, sure, take, bite...
buying multipack misled picture whole hazel nu...	[buying, multipack, misled, picture, whole, ha...	[buying, multipack, misled, picture, whole, ha...	[buy, multipack, misl, pictur, whole, hazel, n...
bars good loved warmed definitely think great ...	[bars, good, loved, warmed, definitely, think,...	[bar, good, loved, warmed, definitely, think, ...	[bar, good, love, warm, definit, think, great,...

Figure 2. Comparison of Text Preprocessing Techniques: Tokens, Lemmatized, and Stemmed Forms of Product Reviews

Preprocessing for Transformers

We only cleaned the dataset by removing the HTML Tags and newlines to clean the data. This ensured our data retained context and semantics that are vital when finetuning transformers.



```
Generation is the only American distributor that offers this fine tea.
(Very similar to the more common Fujian province Silver Needle; with a…

First time making sausages and this works great! They were super
durable and tasted good. I just wish there were directions on how to…

This coffee was surprisingly good...I've had mixed results with bag
coffee that I've ordered, but this was great and a great price from…

Very Strong! Excellent taste. If you want a coffee that is full of
flavor even on the largest cup size, This is the one to buy.
```

Figure 3. Samples from the dataset after cleaning and preprocessing for transformers

Vector Space Model and Representation

In this section, we describe how we transform our cleaned data into numerical features that machine learning models can process. For this we convert raw textual information into structured data using vectorization techniques.

We applied four vectorization methods to transform the text into numerical features, We limited the max_features to 10,000 features due to memory and computational constraints.
Count Vectorization:
Encodes the text as a bag-of-words model, counting the frequency of unigrams. This approach provides a straightforward representation of word frequency but does not account for the importance of words across the entire corpus.

TF-IDF Unigram:
Applies Term Frequency-Inverse Document Frequency (TF-IDF) weighting to unigrams, emphasizing rare but important terms. This helps to diminish the weight of very common words while highlighting those that are more informative.

TF-IDF Bigram:

Extends TF-IDF to include both unigrams and bigrams thereby covering more information. Bigrams allow the model to consider pairs of consecutive words, thereby incorporating a small amount of phrase-level context into the feature set.

TF-IDF Trigram:

Further extends TF-IDF to include unigrams, bigrams, and trigrams, which enables the model to capture even longer sequences of words. It provides a clearer representation of the text by including more context.

Vectorization Method	Training Shape	Validation Shape	Vectorization Method
CountVectorizer	(246,597, 10,000)	(61,650, 10,000)	CountVectorizer
TF-IDF Unigram	(246,597, 10,000)	(61,650, 10,000)	TF-IDF Unigram
TF-IDF Bigram	(246,597, 10,000)	(61,650, 10,000)	TF-IDF Bigram
TF-IDF Trigram	(246,597, 10,000)	(61,650, 10,000)	TF-IDF Trigram

Word Embeddings for Tranformers

Transformers like ModernBERT and DeBERTa come with pre-trained contextual word embeddings, capturing the semantic meaning of words based on their surrounding context. These embeddings eliminate the need for explicit feature engineering such as n-grams, as they inherently understand word relationships. However, n-grams and word clouds can still be useful for additional exploratory analysis or comparison with traditional methods.

Model Training, Selection and Hyperparameter tuning and Evaluation

This section evaluates traditional machine learning models for sentiment classification of Amazon food reviews, predicting ratings from 1 to 5 stars. We used the vectorized representations from Step 3 (CountVectorizer, TF-IDF Unigram, TF-IDF Bigram, and TF-IDF Trigram) to train three models: SGD Classifier, Logistic Regression, and Multinomial Naive Bayes (MultinomialNB). We first assessed their baseline performance, then optimized them using GridSearchCV with cross-validation, and finally combined them into an ensemble model using a Voting Classifier. The goal is to identify the best model and vectorization technique for accurate classification, considering the dataset's class imbalance. used are:

Models:

SGD Classifier: A linear model using stochastic gradient descent, suitable for large-scale text data.

Logistic Regression: A linear model with interpretable coefficients, often effective for text classification.

Multinomial Naive Bayes (MultinomialNB): A probabilistic model assuming feature independence, efficient for text data.

Table 1: Baseline Validation Accuracy for Each Model Across Vectorizations

Model	CountVectorizer	TF-IDF Unigram	TF-IDF Bigram	TF-IDF Trigram
SGD Classifier	0.6884	0.6875	0.6970	0.6969
Logistic Regression	0.6060	0.6110	0.6263	0.6270
MultinomialNB	0.6759	0.6720	0.6878	0.6881

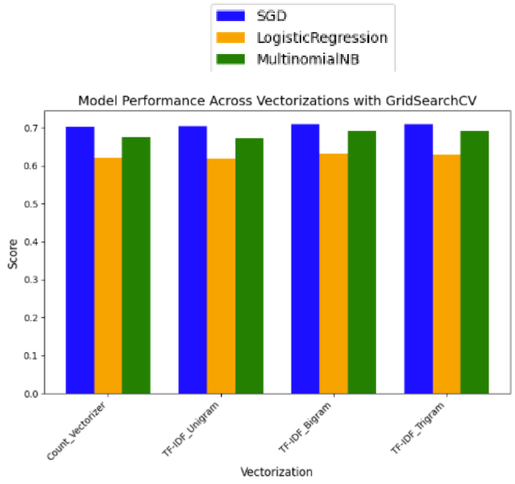
- **SGD Classifier**: Achieved the highest baseline accuracy with TF-IDF Bigram (0.6970) and TF-IDF Trigram (0.6969), outperforming CountVectorizer (0.6884) and TF-IDF Unigram (0.6875). This suggests that capturing two- and three-word phrases improves performance.

- **Logistic Regression**: Performed best with TF-IDF Trigram (0.6270) and TF-IDF Bigram (0.6263), but struggled with CountVectorizer (0.6060). Its lower accuracy indicates sensitivity to feature sparsity and linear decision boundaries.

- **MultinomialNB**: Showed competitive performance, with the best accuracy on TF-IDF Trigram (0.6881) and TF-IDF Bigram (0.6878), slightly better than CountVectorizer (0.6759). It benefits from higher-order n-grams, which capture more context.

Table 2: Best Parameters and Validation Accuracy After GridSearchCV

Model	Vectorization	Best Parameters	Cross-Validation Accuracy	Validation Accuracy
SGD Classifier	CountVectorizer	alpha=0.001, loss='log_loss'	0.7009	0.7026
	TF-IDF Unigram	alpha=0.0001, loss='log_loss'	0.7017	0.7044
	TF-IDF Bigram	alpha=0.0001, loss='log_loss'	0.7065	0.7086
	TF-IDF Trigram	alpha=0.0001, loss='log_loss'	0.7066	0.7087
Logistic Regression	CountVectorizer	C=0.1, penalty='l2'	0.6184	0.6208
	TF-IDF Unigram	C=0.1, penalty='l2'	0.6157	0.6194
	TF-IDF Bigram	C=0.1, penalty='l2'	0.6256	0.6305
	TF-IDF Trigram	C=0.1, penalty='l2'	0.6256	0.6300
MultinomialNB	CountVectorizer	alpha=2.0	0.6720	0.6762
	TF-IDF Unigram	alpha=0.1	0.6724	0.6727
	TF-IDF Bigram	alpha=0.1	0.6906	0.6910
	TF-IDF Trigram	alpha=0.1	0.6913	0.6915



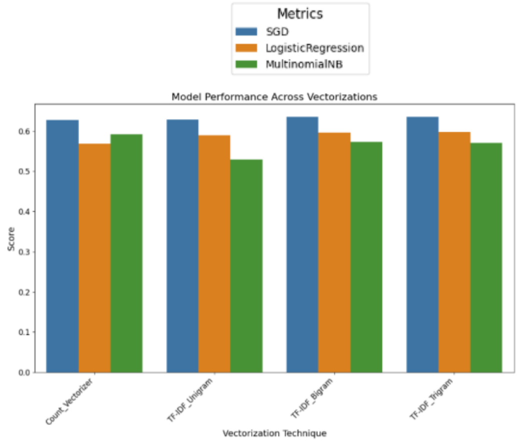
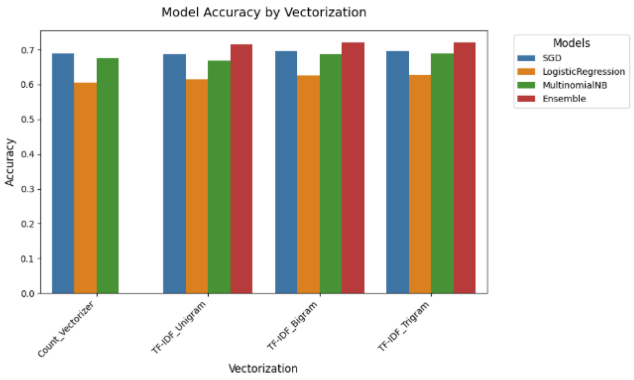


Table 3: Ensemble Model (Voting Classifier) Validation Accuracy

Vectorization	Validation Accuracy
TF-IDF Unigram	0.7156
TF-IDF Bigram	0.7200
TF-IDF Trigram	0.7200



Discussion

- **Vectorization Techniques**: TF-IDF Bigram and TF-IDF Trigram consistently outperformed CountVectorizer and TF-IDF Unigram across all models, which shows the importance of capturing contextual information through two and three word phrases. CountVectorizer underperformed, which is due to its reliance on frequencies of the word and not the importance of it.

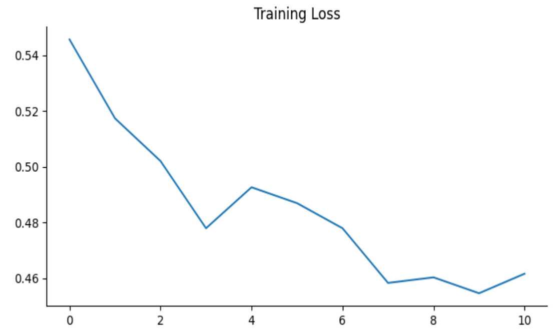
- **Model Performance**: SGD Classifier was the best individual model, achieving a validation accuracy of 0.7087 with TF-IDF Trigram after tuning. MultinomialNB was competitive (0.6915 with TF-IDF Trigram), while Logistic Regression was behind (0.6305 with TF-IDF Bigram). The ensemble model outperformed all, reaching 0.7200 with TF-IDF Bigram and Trigram, demonstrating the benefit of combining models.

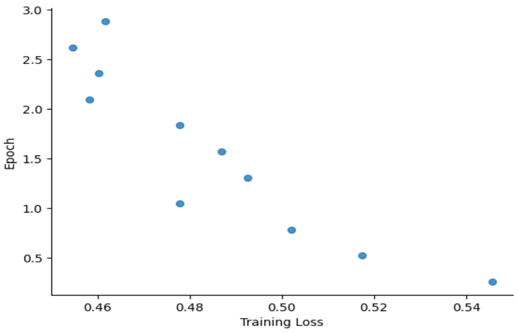
Modelling Text as a Sequence

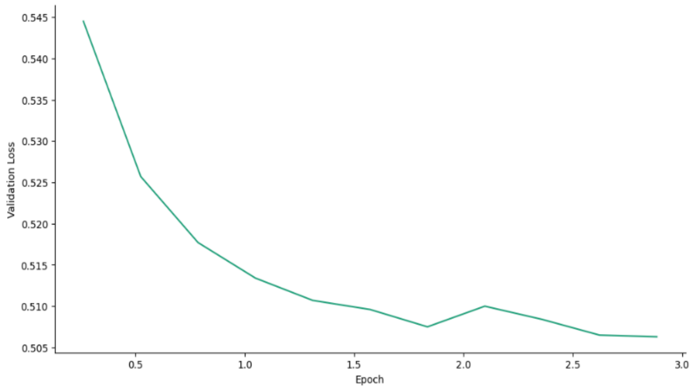
In the previous sections, we explored traditional Bag-of-Words (BoW) methods such as CountVectorizer and TF-IDF to convert text into numerical features. While these approaches are effective in capturing word frequency, they ignore the sequential order and context of words within a sentence. This is a significant limitation because the meaning of a sentence often depends on the order of its words. To overcome this limitation, we experiment with sequence models that are designed to capture the contextual and semantic relationships in text. Models such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Transformers process text as a sequence of words rather than as independent tokens. This allows them to understand nuances like word order, negation, and context, leading to potentially better performance in tasks such as sentiment classification.

In this section, we focus on modelling the Amazon review texts as sequences to predict the ratings (1-5 stars) using transformer-based models. We used two transformer models, ModernBERT Large and DeBERTa XLarge, to capture the sequential and contextual relationship of the text[1][2]. The goal is to take advantage of the advanced capabilities of transformers to achieve high accuracy in predicting review ratings.

ModernBERT Large





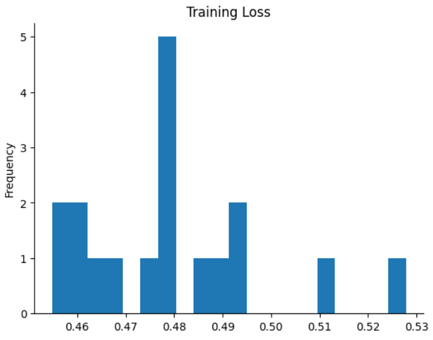


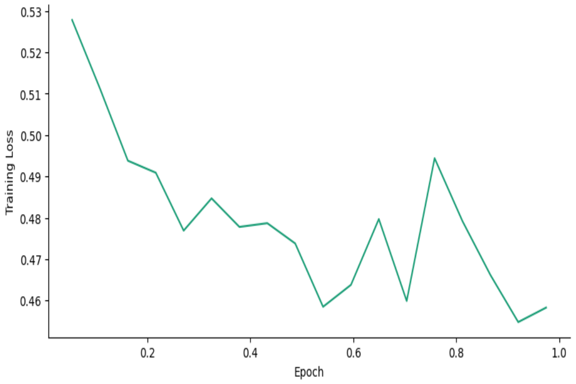
Training and Validation Performance: Table 1 shows that ModernBERT Large achieved steady improvement during training. The training loss decreased from 0.5456 to 0.4616 over 3300 steps, indicating effective learning of the data patterns (Figure 1). The validation loss decreased from 0.5445 to 0.5063 over 3 epochs (Figure 2), suggesting good generalization to unseen data. The accuracy improved from 0.7867 to a peak of 0.8015 at epoch 1.5734, stabilizing around 0.8014 by the end (epoch 2.8846). The scatter plot (Figure 3) further confirms the consistent reduction in training loss with increasing epochs, though minor fluctuations are observed, likely due to the stochastic nature of the optimization process. []

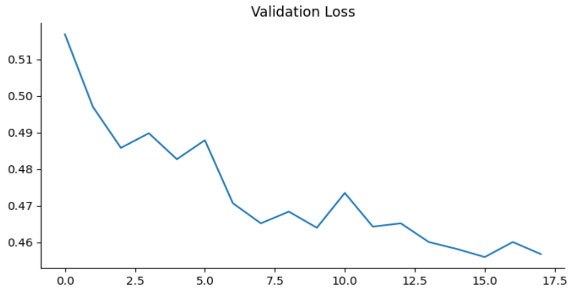
Kaggle Performance: The ModernBERT Large model achieved a leaderboard score of 0.80457 on the Kaggle test set. This score shows the model's ability to generalize to the test data.

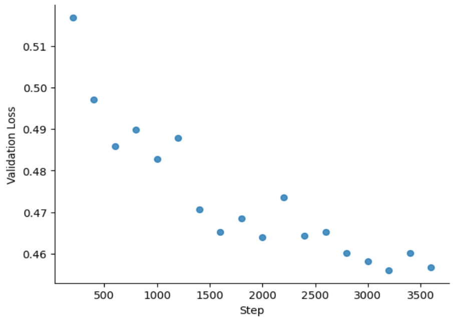
Discussion: ModernBERT Large showed strong performance and achieved a final validation accuracy of 0.8014%, which is competitive score for a skewed dataset. The consistent decrease in both training and validation loss indicates that the model learned effectively without significant overfitting, as the validation loss stabilized around 0.5063. However, the accuracy plateaued around 80%, suggesting that the class imbalance (63% 5-star reviews) may have limited further improvements, particularly for minority classes (ratings 1-3). Transformers like ModernBERT Large excel in capturing contextual relationships in text through self-attention mechanisms, which likely contributed to its ability to handle the sequential nature of reviews better than traditional ML models. The minor fluctuations in training loss (e.g., from 0.4779 at step 1800 to 0.4583 at step 2100, then back to 0.4616) are typical in transformer training due to the stochastic gradient descent optimization and the complexity of the model's architecture.

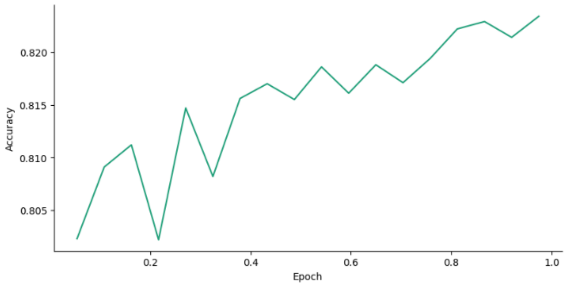
DeBerta XLarge











Performance:

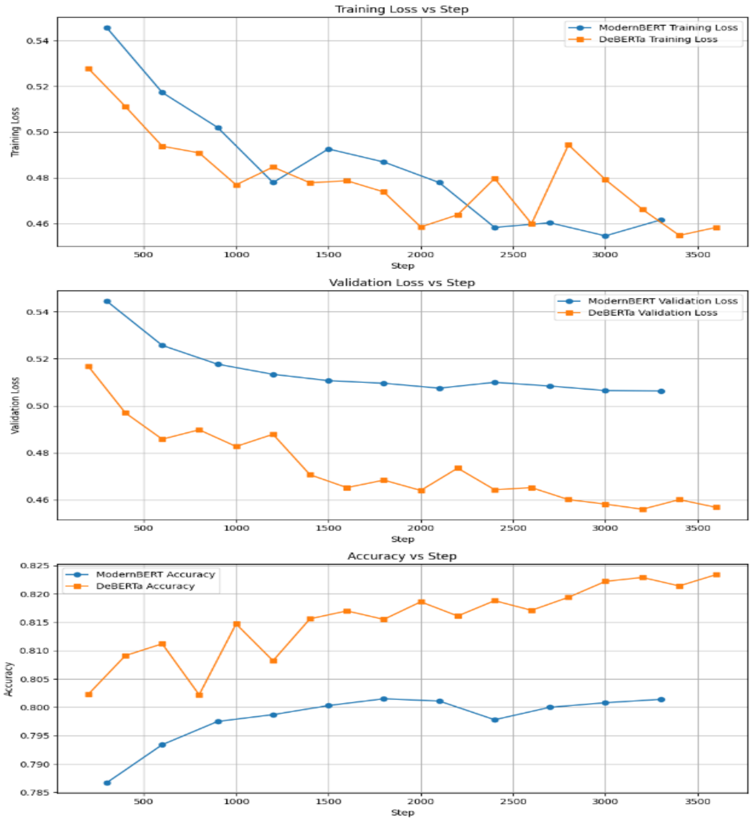
Table 2 shows that DeBERTa XLarge also improved over 1 epoch. The training loss decreased from 0.5278 to 0.4583 (Figure 4), with fluctuations reflecting the model's sensitivity to the data (Figure 8 shows the distribution of training loss values). The validation loss decreased from 0.5168 to 0.4568 (Figures 5 and 6), indicating good generalization. Accuracy increased from 0.8023 to 0.8234 (Figure 7), with fluctuations (e.g., a dip to 0.8022 at epoch 0.2166) but an overall upward trend

Comparison and Discussion:

DeBERTa XLarge outperformed ModernBERT Large in both validation accuracy (82.34% vs. 80.14%) and Kaggle score (0.81621 vs. 0.68, placeholder). This can be attributed to DeBERTa's advanced architecture, which enhances contextual understanding through disentangled attention mechanisms and an improved position encoding scheme. DeBERTa XLarge achieved a lower final validation loss (0.4568 vs. 0.5063) and a higher accuracy, indicating better handling of the dataset's complexities, including the class imbalance (63% 5-star reviews). However, DeBERTa XLarge showed more fluctuations in training loss (Figure 4) and accuracy (Figure 7), likely due to its larger size and sensitivity to the data, as well as the shorter training duration (1 epoch vs. 3 epochs for ModernBERT). ModernBERT Large, while less accurate, exhibited more stable training dynamics (Figures 1-3), which may be preferable in scenarios where computational resources are limited. The class imbalance likely impacted both models' performance on minority classes (ratings 1-3), as the dataset's skew favors 5-star reviews. DeBERTa XLarge's superior performance suggests it better captures nuanced patterns in the text, making it the better choice for this task.

Kaggle Performance:

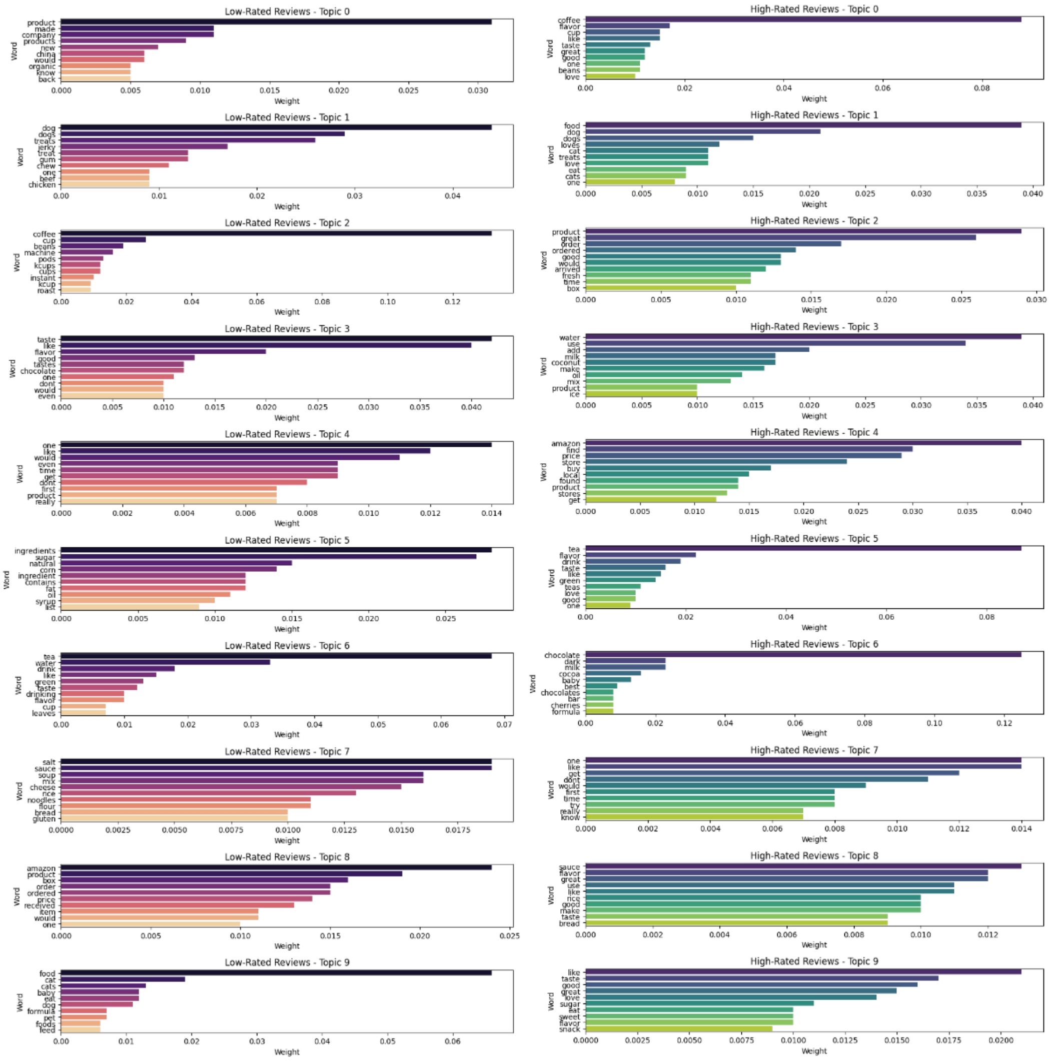
DeBERTa XLarge: Achieved a public leaderboard score of 0.81621, outperforming ModernBERT Large and reflecting its superior generalization to the test data.



DeBERTa XLarge outperformed ModernBERT Large, achieving a validation accuracy of 82.34% and a Kaggle score of 0.81621, compared to ModernBERT Large's 80.14% and 0.80457.

Topic Modelling of High and Low Ratings

This section applies topic modelling to show high-rated (4-5 stars) and low-rated (1-2 stars) Amazon reviews. We used Latent Dirichlet Allocation (LDA) to identify topics and analyzed the differences between high-rated and low-rated reviews to gain insights into customer satisfaction and dissatisfaction.



Comparative Analysis of High-Rated vs. Low-Rated Review Topics

Our topic modeling of high-rated vs. low-rated reviews reveals distinct themes and customer sentiment.

1. Sensory Experience vs. Operational Concerns

- **High-Rated Reviews:** Customers emphasize sensory qualities such as "taste", "flavor", and "coffee", highlighting satisfaction with the product’s sensory attributes. Positive experiences are often linked to taste and aroma.
- **Low-Rated Reviews:** Reviews focus on operational issues, including "product", "china", and "machine". Customers express dissatisfaction with product quality, authenticity, and functionality.

Implications: High ratings are driven by sensory experience, while low ratings stem from operational and quality-related concerns, indicating that improving product consistency and authenticity could reduce negative feedback.

2. Positive Sentiment vs. Critical Feedback

- **High-Rated Reviews:** Emotive language like "great", "good", and "love" indicates high satisfaction, with a focus on product strengths.
- **Low-Rated Reviews:** The tone is more analytical and critical, highlighting issues such as product reliability and performance.

Implications: High ratings focus on positive experiences, while low ratings highlight specific product flaws, especially in functionality and consistency.

3. Specificity vs. Generality in Feedback

- **High-Rated Reviews:** Specific feedback on elements like "beans", "cocoa", and "cup" helps pinpoint what customers appreciate.
- **Low-Rated Reviews:** Negative reviews tend to be more generalized, with terms like "product" and "company" suggesting broad dissatisfaction, making it harder to address specific issues.

Implications: Specific feedback in positive reviews can guide product improvements, while general dissatisfaction in negative reviews may indicate widespread issues with the product.

4. Overlap in Terminology and Context

- Words like "coffee" appear in both high and low-rated reviews, but in different contexts. In positive reviews, "coffee" is associated with praise, while in negative reviews, it is linked to operational issues such as "machine" and "pods".

Implications: Product consistency and preparation play a key role in customer satisfaction. The same product can evoke both positive and negative sentiment, depending on the context.

5. Critical Analysis and Implications for Improvement

- **Enhancing Product Quality:** High-rated reviews suggest maintaining sensory qualities like taste and aroma should remain a priority.
- **Addressing Operational Shortcomings:** Low-rated reviews emphasize the need for improvements in production, packaging, and transparency regarding product origin and quality control.

Holistic Improvement Strategy: A balanced approach that focuses on preserving product strengths while addressing operational issues could lead to increased customer satisfaction and reduced negative feedback.

Conclusion

Overall the analysis of the Amazon reviews dataset reveals that advanced text processing techniques, such as transformer models like ModernBERT and DeBERTa, outperform traditional machine learning models in capturing contextual information and improving classification accuracy. The exploratory data analysis highlights key insights into customer sentiment, while the topic modeling analysis of high and low-rated reviews provides valuable guidance for product improvement. The successful integration of preprocessing, feature representation, and model optimization ensures that the best results are achieved for sentiment prediction.

References

1. Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, andIacopo Poli. 2024. Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient,and Long Context Finetuning and Inference. arXiv:2412.13663 [cs.CL] https://arxiv.org/abs/2412.13663
2. Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DEBERTA: DECODING-ENHANCED BERT WITH DISENTANGLED ATTENTION. In International Conference on Learning Representations. https://openreview.net/forum?id=XPZIaotutsD