School of Physics,
Engineering and
Computer Science

MSc Data Science Project
7PAM2002
Department of Physics, Astronomy and Mathematics

Data Science FINAL PROJECT REPORT

Project Title:

Sentiment Analysis of Amazon Reviews Using Machine Learning and Transformer Models

Student Name and SRN:

Muhammad Saad Bin Sagheer

23086746

Supervisor: Klaas Wiersema

Date Submitted: 6 January 2026

Word Count: 4920

GitHub Link: https://github.com/Saadi222/MSc-Final-Project.git

DECLARATION STATEMENT

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science **in Data Science** at the University of Hertfordshire.

I have read the detailed guidance to students on academic integrity, misconduct and plagiarism information at Assessment Offences and Academic Misconduct and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project or course.

I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6)

I did not use human participants in my MSc Project.

I hereby give permission for the report to be made available on module websites provided the source is acknowledged.

Student Name printed: Muhammad Saad Bin Sagheer

Student Name signature

Student SRN number: 23086746

UNIVERSITY OF HERTFORDSHIRE

SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

ABSTRACT

This project analyses Amazon product reviews using Natural Language Processing (NLP) techniques to automatically classify customer sentiment. The project investigates whether transformer-based models, specifically DistilBERT, outperform classical machine learning algorithms such as Logistic Regression and Random Forest when applied to noisy textual data. The dataset comprises publicly available Amazon customer reviews that were pre-processed using cleaning, tokenisation, TF-IDF representation, and class-imbalance handling. Exploratory data analysis was conducted to examine rating distributions, review patterns, and linguistic characteristics. Sentiment classification models were trained and evaluated using accuracy and F1-scores. The project concludes by comparing model behaviours, discussing the strengths of transformer architectures in NLP, and outlining real-world applications for organisations seeking to automate large-scale review analysis.

Table of Contents

List of Figures:

List of Tables:

Introduction

Online product reviews constitute an essential source of information in digital marketplaces. Customers on platforms like Amazon, seeking to make informed choices, heavily depend on the reviews and ratings that they come across. On the other hand, companies employ this feedback to evaluate the quality of their products, keep track of customer satisfaction, and plan product development. The sheer number of reviews that are being posted every day makes it impossible for anyone to analyse them manually and hence, automated methods must be used to scale up the process. As a result, the use of Natural Language Processing (NLP) and machine learning has become the mainstay in the industry to identify consumer opinions, detect sentiment, and summarise feedback to facilitate decision making. Sentiment analysis has been the primary focus of various traditional machine learning techniques including Logistic Regression and Random Forests, which usually employ bag of words or TF-IDF representations. Lately, transformer-based models such as BERT and DistilBERT have set new standards for performance by understanding the context of the text. The main issues that the researchers are still grappling with are the noisy user-generated text, class imbalance in the datasets, and model generalisability over different product categories.

Research Question

To what extent can machine learning and transformer-based models accurately classify sentiment in Amazon product reviews, and how do their performances compare when evaluated on the same dataset?

Aims and Objectives

The main aim of this project is to build and evaluate sentiment classification models from Amazon dataset, and to compare the performance of classical machine learning algorithms with that of a transformer model (DistilBERT). The main objectives are as follows:

1. To perform exploratory data analysis (EDA) to understand review distributions, text characteristics, and sentiment patterns.
2. To clean and preprocess review text using standard NLP techniques such as tokenisation, stopword removal, and lemmatisation.
3. To implement baseline machine learning models (Logistic Regression, Random Forest) using TF-IDF features.
4. To fine-tune a DistilBERT transformer model for sentiment classification.
5. To perform tuning on models to improve performance and address class imbalance.
6. To evaluate and compare model results using accuracy, macro F1-score, weighted F1-score, and confusion matrices.
7. To interpret results and identify which modelling approach is most effective for sentiment classification of Amazon reviews.

Literature Review

I have used 3 different types of models in this project.

1. Logistic Regression

Logistic Regression has been widely adopted as a baseline model for sentiment analysis due to its computational effectiveness and strong performance on high-dimensional sparse text representations. Pang and Lee (2008) demonstrated that linear classifiers, when combined with bag-of-words features, can effectively classify sentiment settings, despite being originally designed for topic classification. Their work showed that sentiment classification poses unique challenges, as sentiment often depends on subtle linguistic hints rather than dominant topic words.

Joachims (2002) further established the suitability of linear models for text classification by showing that they scale efficiently with large feature spaces generated by TF-IDF representations. Logistic Regression benefits from this property and often generalises well when regularisation is applied. However, a key limitation highlighted in the literature is that Logistic Regression ignores word order and semantic context, treating text as an unordered set of features. As a result, it struggles with negation, sarcasm, and changes in contextual polarity. In this project, Logistic Regression is used as a strong and interpretable baseline against which more complex models are evaluated.

2. Random Forest

Random Forest is an ensemble machine learning algorithm that aggregates predictions from multiple decision trees to improve robustness and reduce variance. Kowsari et al. (2019) analysed Random Forest performance across multiple text classification tasks and reported that while ensemble methods can capture non-linear patterns, they often perform poorly on sparse, high-dimensional feature spaces such as TF-IDF matrices. The research emphasised that decision trees tend to memorise training examples, resulting in overfitting when applied to textual data.

This limitation is especially troublesome for sentiment analysis datasets that are imbalanced, where dominant sentiment classes can bias tree splits. While Random Forest may be able to beat linear models in some structured datasets, it is still mostly ineffective in NLP tasks without dimensionality reduction or feature selection. The work of Kowsari et al. is very similar to what we see here that Random Forest gets very high training accuracy but lower validation accuracy, thus overfitting, in their study as well. Despite that, the Random Forest still serves as a valuable reference point for comparison, as it is a fundamentally different modelling approach from linear classifiers.

3. Transformer-Based Models (DistilBERT)

Transformer architectures have revolutionized sentiment analysis by allowing models to understand the context of the text. BERT, introduced by Devlin et al. (2019), is a model that uses bidirectional self-attention to understand the meaning of a word based on the context. Their paper showed that BERT achieved very large improvements in performance over traditional machine learning models for various NLP tasks including sentiment classification. In contrast to TF- IDF based models, BERT handles semantic relations, negations, and long dependencies.

Nevertheless, BERT is a very large model and requires a lot of computation, hence it is not a very suitable model for some applications. DistilBERT, a smaller version of BERT obtained by knowledge distillation, was proposed by Sanh et al. (2019). Their experiments indicated that DistilBERT keeps about 97% of BERT's performance while it is considerably smaller and faster. Thus, DistilBERT is a very good choice for sentiment analysis in real life scenarios where computer efficiency is a key factor.

Dataset Description

The data for this project is customer review data scraped from the Amazon e-commerce platform. The data was published on Kaggle, a popular online platform for data science datasets and competitions.

Each entry in the dataset corresponds to one customer review and contains both textual and metadata attributes. These include review text, review title, star rating (1 to 5), review date, and some reviewer information. In this project, the review text is the main input feature for sentiment classification models. The sentiment labels are star ratings: 1 and 2 correspond to negative sentiment, 3 to neutral sentiment, and 4 and 5 to positive sentiment.

Data Pre-processing and Feature Engineering

The CSV file held data needed a lot of work to be done before it could be used for sentiment analysis. The very first thing was to load the dataset and find the missing values, duplicate records and also the inconsistent formatting. To make column names uniform throughout the processing steps, whitespaces were removed from them.

The difficult thing about the dataset was the inconsistency in the representation of rating information, some entries looked like textual strings instead of numeric values. To fix this problem, a custom parsing function was written to get the numerical ratings from the rating column. Later, these numeric ratings were used as a basis for the creation of sentiment labels, thus categorising reviews into negative, neutral, or positive ones. The reviews that had missing or invalid rating values were not included in the supervised model training.

The textual review data was heavily contaminated with noise due to the nature of user, generated content. A thorough text preprocessing pipeline was carried out, which involved converting the text to lowercase, stripping HTML tags, URLs, non, ASCII characters, punctuation, and excessive whitespace. Common English stopwords were removed to lessen the noise, and lemmatisation was performed to normalise word forms. The result was a cleaned version of the review text, which was saved in a different column and considered the main input for all models. Duplicated reviews were detected and removed through a combination of reviewer identifiers and review text to avoid data leakage and biased learning. Rows with missing review text were also dropped as they contain no useful information for sentiment classification. To support exploratory data analysis and feature engineering, more numerical features were created from the cleaned text such as word count, character count, number of unique words, and average word length. These features made it possible to study distributions of review lengths and their correlation with sentiment polarity.

In the end, classical machine learning models like Logistic Regression and Random Forest utilized the cleaned text data, which was converted into numerical forms through TF-IDF vectorisation. This resulted in high dimensional sparse feature matrices that were appropriate for linear and ensemble classifiers. The cleaned text for the DistilBERT model was tokenised with a pre-trained tokeniser, thus making it consistent with the transformer architecture.

In summary, the data pre-processing pipeline made sure that the dataset was clean, consistent, and of the right format for both traditional machine learning and deep learning models. Moreover, it retained the semantic content that is essential for correct sentiment analysis.

Exploratory Data Analysis

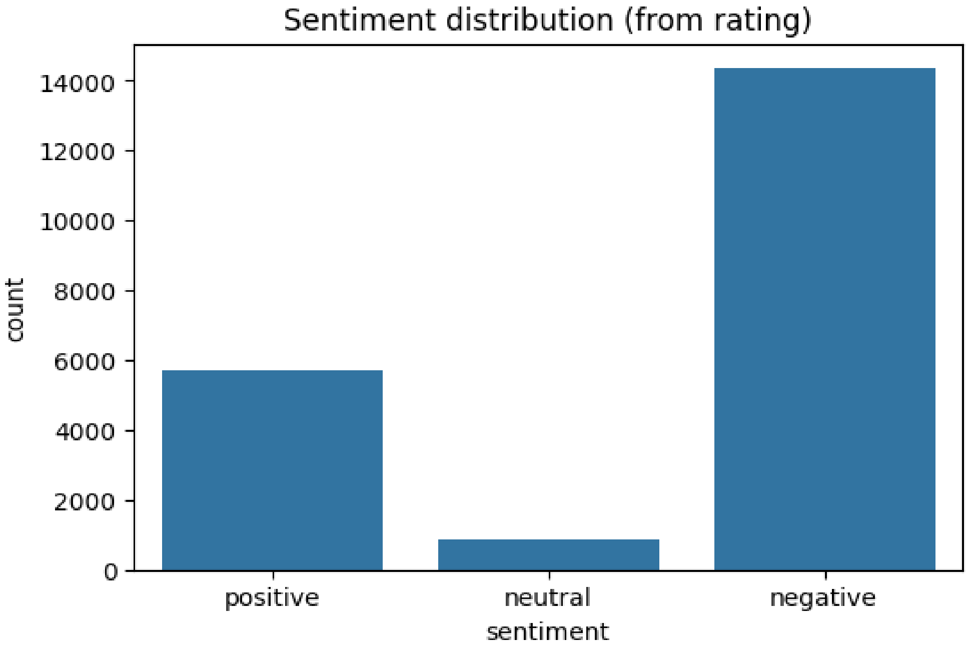


Figure 1: Distribution of Sentiment Classes

Figure 1 presents the proportions of sentiment classes labelled based on review ratings. The figure reveals that the distribution of sentiment classes is heavily imbalanced, with negative reviews being the largest group of observations, followed by positive reviews, and neutral reviews accounting for a very small fraction of the data.

The class imbalance implies that customers are more inclined to write reviews after negative experiences, which is a typical occurrence in online feedback platforms. The small number of neutral reviews indicates that users are more likely to express strong opinions rather than provide moderate feedback. Such skewed class distributions pose difficulties for supervised learning models as classifiers can become biased towards the majority class and achieve high accuracy scores that are misleading, while at the same time, they fail to recognize the minority sentiment categories correctly.

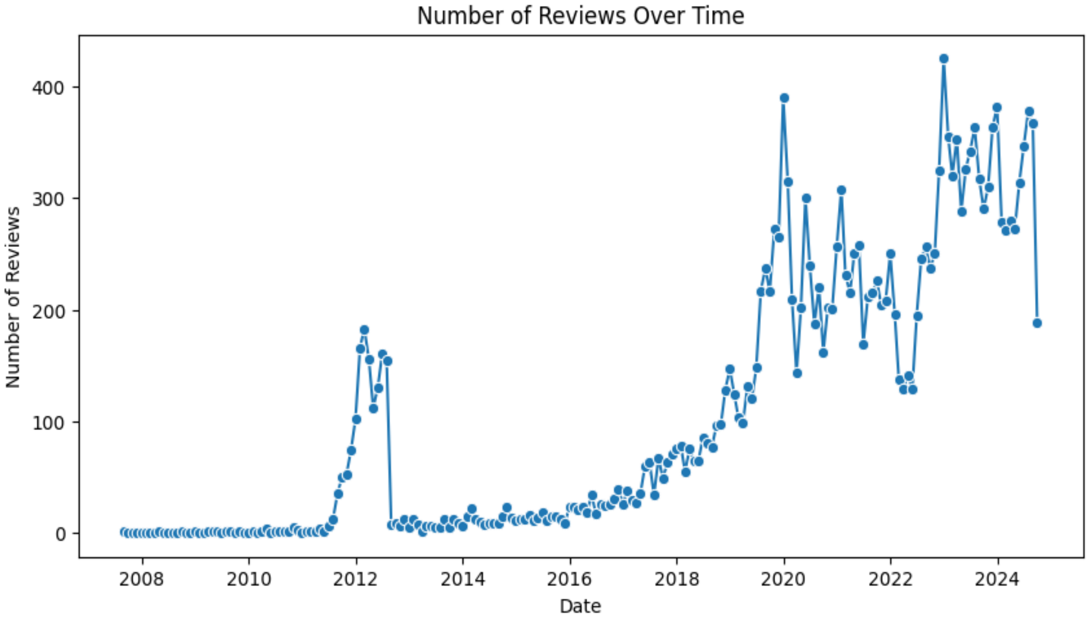


Figure 2: Number of reviews over time

Figure 2 presents the volume of reviews submitted over the years. Review activity has clearly increased over time, especially after 2017, as the number of users on the platform has risen. Very few reviews can be seen in the first years, which also suggests that platform was less used during those times.

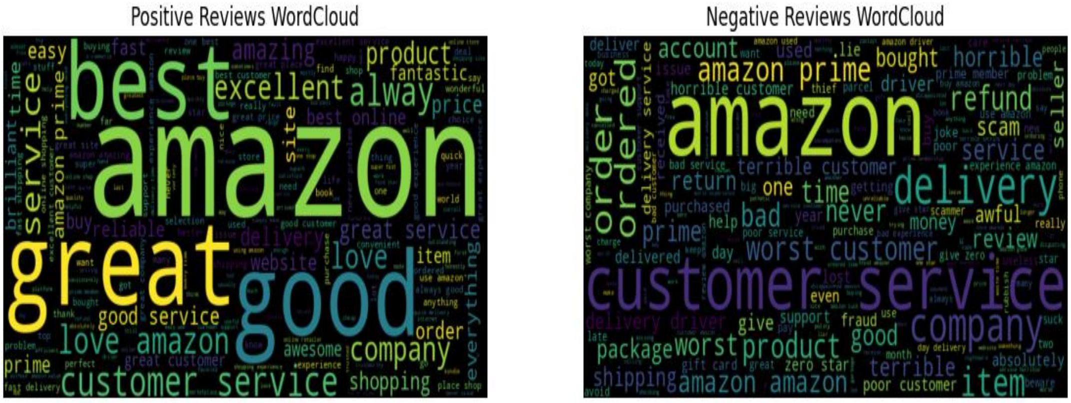


Figure 3: Word Cloud for positive and negative reviews

Positive and negative reviews word clouds are shown in Figure 3. Words like "good", "great", "excellent", "love", and "best" are frequently used in positive reviews, which indicate that customers are generally satisfied and have had a good experience with the product. These words are frequently used to describe product quality, value, and service reliability.

On the other hand, negative reviews contain words like "bad", "worst", "problem", "refund", "delivery", and "customer service" which indicate that the dissatisfaction is due to product defects, delayed delivery, and service, related issues. The appearance of transactional words like "refund" and "order" implies that the negative sentiment is mostly about the post-purchase experience and not the product.

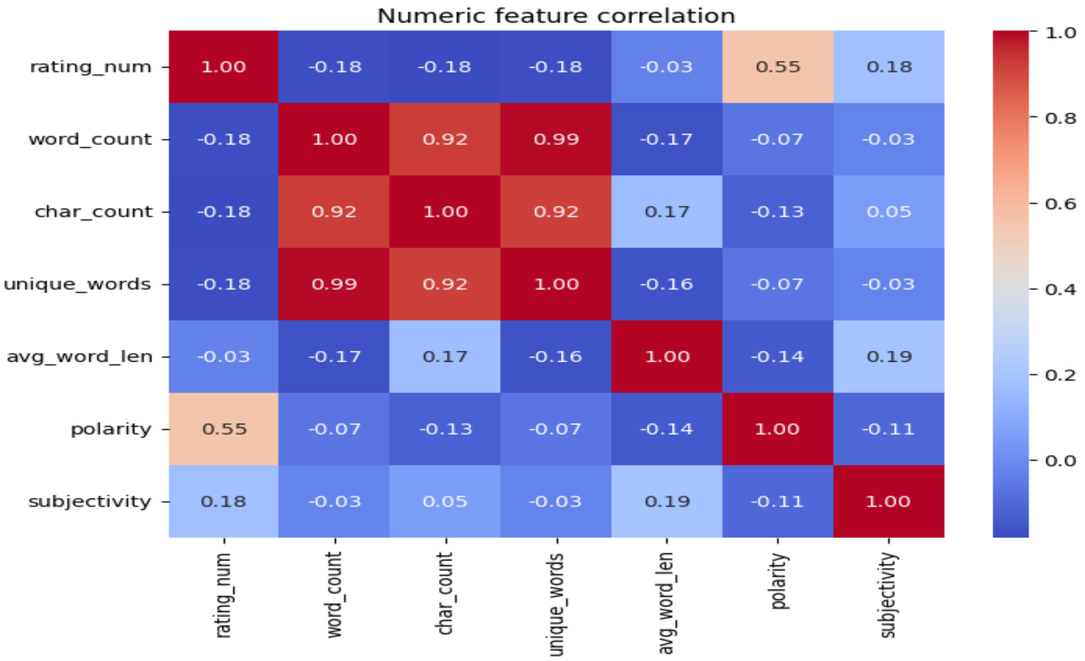


Figure 4 Correlation matrix of numeric features

Figure 4 illustrates the correlation of the numeric features derived from the dataset. The features describing review length, such as word count, character count, and number of unique words, experienced a remarkably high correlation with each other. The positive correlation between rating and sentiment polarity is moderate, which means that numerical ratings and textual sentiment generally agree, but not always. Most of the remaining features have very weak correlations, which means that sentiment can be more accurately extracted from the text than from the numeric attributes.

Model Selection and Evaluation Metrics

Based on the literature review and project objectives, I selected three models for classifying Amazon review sentiment. These three models are Logistic Regression, Random Forest, and DistilBERT. Their selection is motivated by the fact that they represent three different modelling paradigms: machine learning, ensemble learning, and transformer-based deep learning. I split the dataset into training and test subsets with an 80-20 split, which means 80% of the data was used for training, and 20% for testing. The training set was used to fit the models, while the test set was kept aside for evaluating performance on unseen data to assess generalisation. Because of the imbalance in the dataset, accuracy was not enough to evaluate the model's performance. Hence, multiple classification metrics were used to ensure a comprehensive assessment, including accuracy, precision, recall, and F1-score. F1-score was considered as the most important measure since it gives a balanced view of the model's effectiveness by considering both false positives and false negatives, thus being a suitable metric for imbalanced classification problems.

The evaluation framework made it possible to compare the selected models on various criteria besides prediction. Specifically, the model's generalisation capacity to new data was also measured, thus enabling an accurate comparison between traditional machine learning approaches and transformer-based models.

Accuracy

It calculates the proportion of correct classifications over all the predictions that the model has made. It represents the overall performance of the model; nevertheless, it may be deceptive when there is a class imbalance situation because a model can achieve a high accuracy score just by predicting the majority class.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

- TP= True Positives
- TN= True Negatives
- FP= False Positives
- FN= False Negatives

F1-Score

The F1- score is the harmonic mean of precision and recall and thus represents a balanced evaluation of a model's performance, especially in the case of imbalanced classification problems. If a model performs well on one metric but badly on the other, the F1 score will still be low, hence it is more reliable than accuracy in such cases.

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where:

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

A weighted F1-score was used to account for class imbalance by weighting each class's F1-score according to its support (number of samples). This technique confirms that the evaluation reflects model performance across all sentiment classes rather than being dominated by the majority class.

For evaluation to reflect the performance of the model in all sentiment categories rather than be skewed to the majority class, a weighted F-1 score was used.

Logistic Regression

I chose Logistic Regression as a first model for sentiment classification because of its simplicity, high interpretability, and good performance on high-dimensional text data. In NLP tasks, Logistic Regression is typically used along with TF-IDF feature representations, so it is a good candidate to use as a standard model and compare more complicated methods like ensemble models and transformers to it.

Logistic Regression is one of the linear models for classification, and it tries to find the different probabilities of a given instance belonging to a certain class using the logistic (sigmoid) function. Unlike regression models that predict a continuous value, Logistic Regression outputs class probabilities, which are then mapped to sentiment classes.

Model Description

Given an input feature vector x, Logistic Regression models the probability of class membership as:

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-(\beta_0 + \beta^T x)}}$$

where:

- $\beta_0$is the bias term
- βrepresents the model coefficients
- x is the TF-IDF feature vector

Feature Representation

The cleaned review text was converted to a numeric term frequency-inverse document frequency (TF-IDF) vector. TF-IDF uses a higher weight on words that are common in a review but uncommon across the entire dataset to allow the model to concentrate on terms with sentiment-bearing information instead of concentrating on common words. The unigrams and bigrams were added to represent the short contextual phrases like the negations (e.g. not good) that are significant in sentiment classification. This led to a high-dimensional, sparse feature space, which could be used with linear classifiers

Handling Class Imbalance

Since Exploratory Data Analysis revealed that there is a significant unequal distribution of the sentiment classes, with negative reviews prevailing in the dataset. To deal with this problem, random oversampling was used so we could have an equal representation of each sentiment class in the training of the model. This helped to make sure that the Logistic Regression model was not biased towards the majority category (negative) and made it better at correctly classifying minority sentiment categories.

Model Results

The initial Logistic Regression model achieved an overall accuracy of 85.8% and the weighted F1-score of 0.87 on the test data set. The model performed very well on the negative and positive sentiment categories, achieving F1 scores of 0.93 and 0.83, respectively, indicating successful classification of most sentiment categories. The performance of the neutral class was, however, low with the F1-score of 0.17, which was heavily contributed to by the few samples of neutral sentiment classes. The F1-score of 0.64 with macro-averaging highlights the impact of class imbalance, as the performance of the minority class significantly reduced the macro-averaging score.

To enhance the initial baseline Logistic Regression model, tuning was conducted using GridSearchCV. The choice of the solver and the regularisation strength parameter were optimised using five-fold cross-validation, with the weighted F1-score used as the assessment metric to account for class imbalance. Eight parameter sets were tested across 40 model fits. It was observed that the most suitable option was L2 regularisation with the lbfgs solver. The tuned Logistic Regression performed better than the untuned baseline model, especially for the dominant sentiment classes. The model achieved the 86% accuracy with a weighted F1-score of 0.87 on the test set. As performance in the neutral class was relatively low due to the small number of samples, the tuned model was found to be more generalised and class-balanced, which validates the usefulness of regularisation optimisation in very high-dimensional contexts of text classification.

Model Results and Comparison

Metric	Baseline Logistic Regression	Tuned Logistic Regression
Accuracy	0.858	0.860
Weighted F1-score	0.869	0.870
Macro F1-score	0.64	0.64
Negative F1-score	0.93	0.93
Neutral F1-score	0.17	0.14
Positive F1-score	0.83	0.83
Regularisation (C)	Default(C=1)	Optimised(C=10)
Solver	Default	lbfgs
Class Weighting	Balanced	Balanced

Table 1: Comparison before and after tuning of Logistic Regression

Based on Table 1 comparisons, it is apparent that hyperparameter tuning did not yield significant gains in overall accuracy or weighted F1-score, indicating it did not improve over the baseline model. The performance of most sentiment classes (negative and positive) was not significantly different, whereas the neutral class remained poor. The general results show that Logistic Regression is already performing nearly optimally on this data.

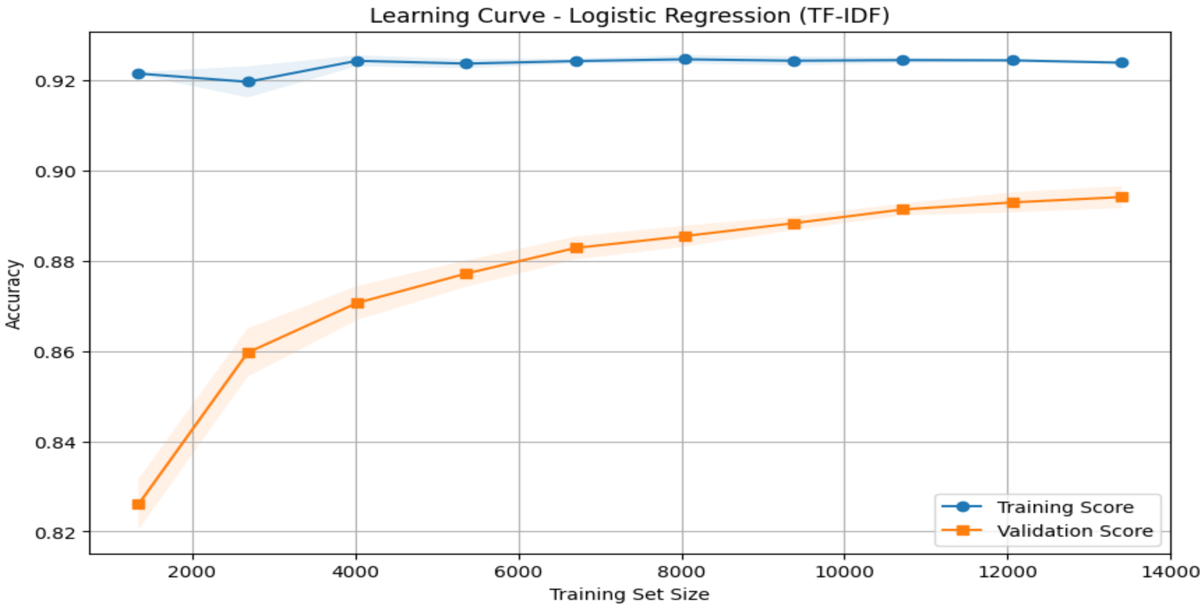


Figure 5: Correlation matrix of numeric features

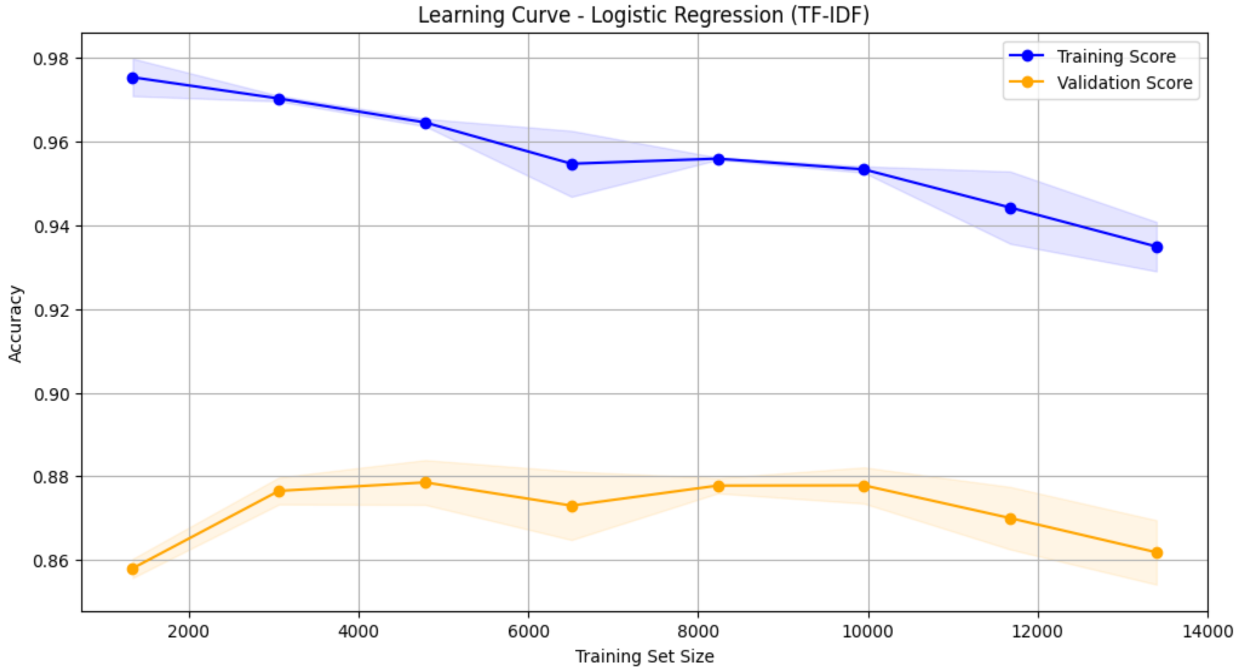


Figure 6 Learning curve for tuned Logistic Regression

Figures 5 and 6 indicate the learning rates of baseline and tuned logistic regression, respectively. The learning curves show that there are no significant differences between the baseline and tuned Logistic Regression models, which have high training accuracy and a low but consistent difference between training and validation accuracy. Following hyperparameter tuning, training accuracy drops marginally, yet validation accuracy does not vary greatly, depicting higher regularisation and less overfitting. Nonetheless, there is still a gap between training and validation performance in both cases, and tuning appears to result in greater stability in generalisation rather than a significant improvement in performance. In general, the learning curves indicate that hyperparameter optimisation has only a slight impact on the model's strength, as measured by Logistic Regression on this dataset

Random Forest

The ensemble model I used is Random Forest. It was used to assess whether a non-linear ensemble learning approach could improve sentiment classification performance in comparison with linear models. Unlike Logistic Regression, which assumes a linear relationship between features and class labels, Random Forests can capture complex, non-linear feature interactions, making them highly appropriate for high-complexity text data (Breiman, 2001)

Model Description

Random Forest is an ensemble learning algorithm constructed from multiple decision trees trained on bootstrapped samples of the training data and randomly selected feature subsets. The majority voting over all trees is used to obtain the final prediction:

$$\hat{y} = \text{mode}\{h_1(x), h_2(x), \ldots, h_T(x)\}$$

Where:

- $h_t(x)$is the prediction of the t^{th}decision tree
- $T$is the total number of trees in the ensemble

This ensemble method reduces variance and controls overfitting compared to a single decision tree, thereby improving stability and robustness (Breiman, 2001).

Feature Representation

The same TF-IDF representation, train–test split strategy, and class imbalance handling techniques described in the Logistic Regression section were applied to provide a fair model comparison.

Model Training

The Random Forest model was trained with a fixed number of trees, allowing each tree to grow fully without depth constraints. While this configuration enables the model to learn rich decision boundaries, it also increases the likelihood of overfitting when applied to sparse and high-dimensional TF-IDF features.

Model Results

The Random Forest achieved 85% accuracy and a weighted F1-score of 0.85 on the test dataset. The model was effective in most of the sentiment classes with an F1 of 0.92 on negative reviews and 0.79 on positive review indicating that the model was effective in classifying the most dominant sentiment classes. Nevertheless, the neutral class score was significantly low, with an F1-score of 0.09, indicating not only extreme imbalance among the classes but also an inability to differentiate neutral sentiment from the other classes. The macro-average F1 score of 0.60 also emphasises the model's poor performance with respect to minority classes. Despite the high fitting power, the overall classification performance of the Random Forest did not outperform that of Logistic Regression, indicating that more sophisticated models did not necessarily result in higher sentiment classification performance on sparse textual data.

Hyperparameter tuning for the Random Forest model was conducted using Randomised Search with three-fold cross-validation, evaluating twenty parameter combinations. The optimal configuration included 234 trees with unrestricted depth, square-root feature selection, and a minimum of 4 samples per node. Despite this optimisation, the tuned Random Forest achieved an accuracy of 84.9% and a weighted F1-score of 0.84, which represents a slight decrease compared to the untuned baseline model. Performance for the negative and positive sentiment classes remained largely stable, while classification of the neutral class deteriorated further, with an F1-score of 0.07. The macro-averaged F1-score of 0.59 highlights persistent difficulty in generalising across minority classes. These results indicate that hyperparameter tuning did not substantially improve model generalisation and further suggest that Random Forest may not be well-suited to sparse TF-IDF representations for sentiment classification in this dataset.

Model Results and Comparison

Metric	Baseline Random Forest	Tuned Random Forest
Accuracy	0.85	0.85
Weighted F1-score	0.85	0.843
Macro F1-score	0.60	0.59
Negative F1-score	0.92	0.92
Neutral F1-score	0.09	0.07
Positive F1-score	0.79	0.78
Number of Trees	200	234
Feature Selection	Default	Sqrt
Hyperparameter Tuning	No	Yes (CV = 3)

Table 2: Comparison before and after tuning of Random Forest

From Table 2, we can conclude that tuning led to a slight decrease in both accuracy and weighted F1-score, indicating no improvement over the baseline model. Performance for the majority sentiment classes remained largely unchanged, while classification of the neutral class deteriorated further. These results suggest that Random Forest is prone to overfitting when applied to sparse TF-IDF representations and that hyperparameter tuning alone cannot overcome this limitation. Overall, increasing model complexity did not improve performance.

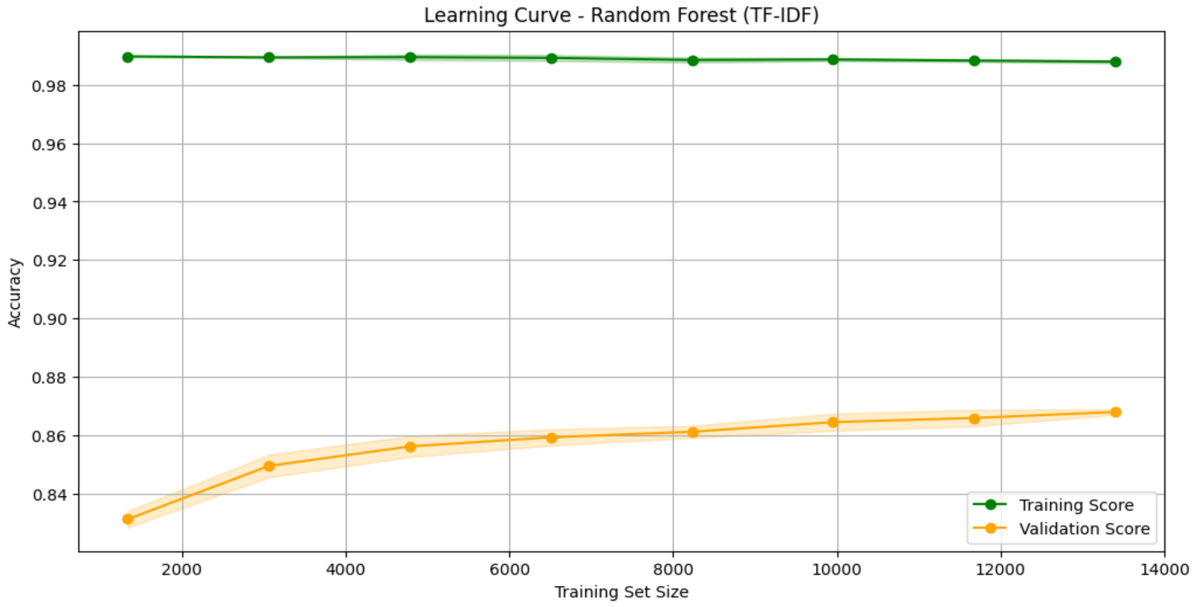


Figure 7: Learning curve for baseline Random Forest

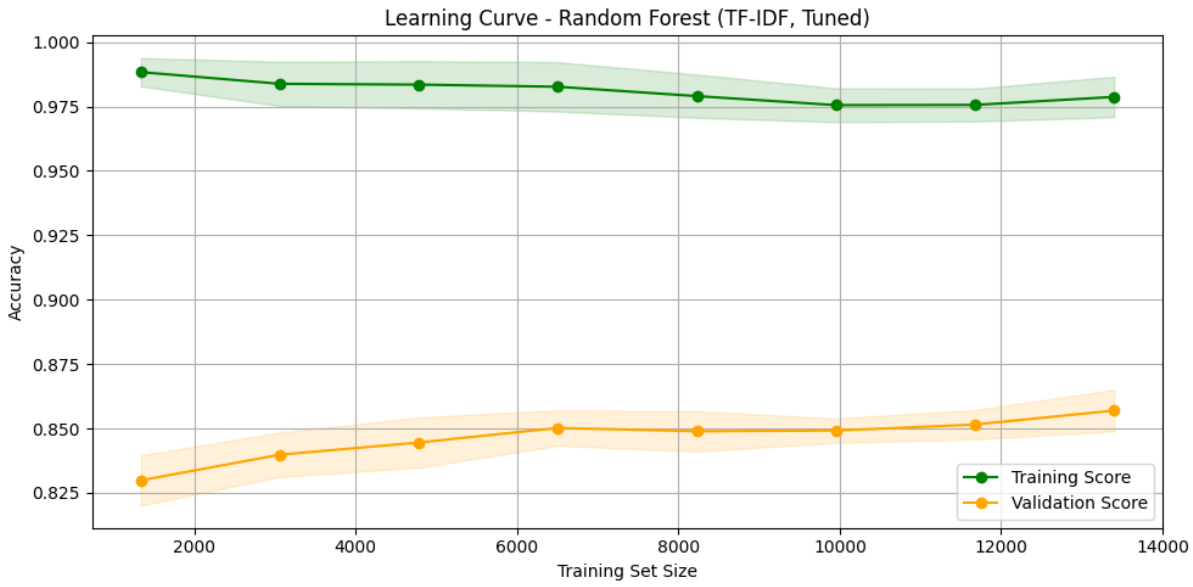


Figure 8: Learning Curve for tuned Random Forest

The learning curves from figures 7 and 8 indicate that both the baseline and tuned Random Forest models exhibit a large and persistent gap between training and validation accuracy, confirming strong overfitting. In the untuned model, training accuracy remains close to 99% across all training sizes, while validation accuracy improves only marginally as more data is added. After hyperparameter tuning, training accuracy decreases slightly, indicating reduced model complexity; however, the validation accuracy does not improve meaningfully and remains lower than the baseline model. This suggests that although tuning marginally constrains the model, it does not resolve the overfitting problem. Overall, hyperparameter tuning reduces training performance without improving generalisation, reinforcing the idea that Random Forest struggles to generalise effectively on sparse TF-IDF text features.

DistilBERT

DistilBERT was selected as the deep learning model for this project to evaluate whether a transformer-based architecture can improve sentiment classification performance by learning contextual representations of text. Unlike Logistic Regression and Random Forest, which rely on TF-IDF features, DistilBERT processes raw text directly and captures semantic meaning using self-attention mechanisms (Sanh *et al.*, 2019)

Model Description

DistilBERT is a compressed version of BERT that retains approximately 97% of BERT's language understanding capability while being significantly faster and lighter. The model consists of stacked transformer encoder layers that use self-attention to model contextual relationships between words in a review.

For sentiment classification, the final hidden representation corresponding to the special [CLS] token is passed through a feed-forward classification head:

$$\hat{y} = \mathrm{softmax}(W h_{[\mathrm{CLS}]} + b)$$

Where:

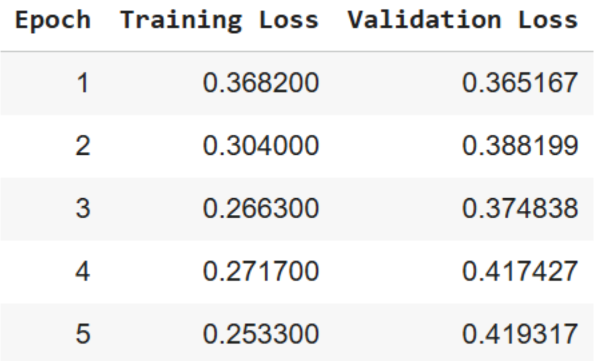
- $h_{[\mathrm{CLS}]}$ is the contextual embedding of the input sequence

- $W$and $b$are trainable parameters
- $\hat{y}$ represents the predicted class probabilities

Model Training

The same train–test split and sentiment labels described earlier were used. Tokenisation was performed using the DistilBERT tokeniser, which converts text into subword tokens and applies padding and truncation to a fixed sequence length. Class imbalance was addressed through weighted loss, making sure that minority classes contributed proportionally to the training objective

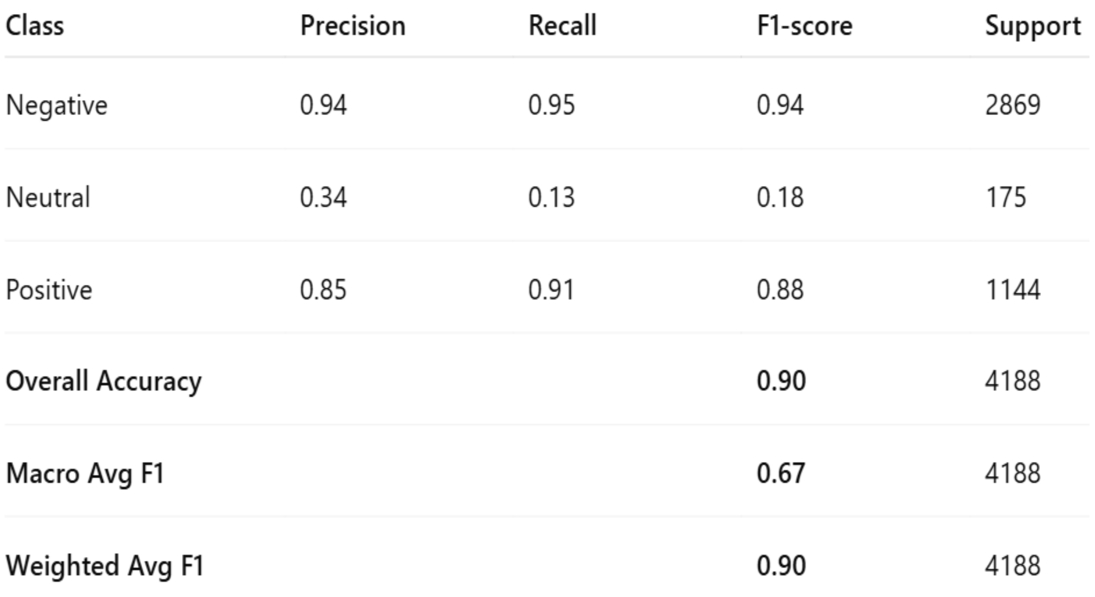
Model Results



Epoch	Training Loss	Validation Loss
1	0.368200	0.365167
2	0.304000	0.388199
3	0.266300	0.374838
4	0.271700	0.417427
5	0.253300	0.419317

Table 3: Training and Validation Loss for DistilBERT

The table shows the training and validation loss of the DistilBERT model across five training epochs. Over the course of the training, the loss steadily reduced from 0.368 to 0.253, showing that the model is indeed learning and fitting the training data well. Meanwhile, the validation loss increased after the first epoch, rising from 0.365 to 0.419 at the fifth epoch. Such a divergence between training and validation loss isa strong signal of overfitting, i.e. The model starts to recall the training data rather than generalising to unseen reviews. The findings imply that optimal performance is attained in the first few epochs only and that further training deteriorates hemodel's ability to generalise.



Class	Precision	Recall	F1-score	Support
Negative	0.94	0.95	0.94	2869
Neutral	0.34	0.13	0.18	175
Positive	0.85	0.91	0.88	1144
Overall Accuracy			**0.90**	4188
Macro Avg F1			**0.67**	4188
Weighted Avg F1			**0.90**	4188

Table 4: Results of DistilBERT

The DistilBERT model was the best performer in terms of overall metrics among all the models tested. The model yielded an accuracy of 90.4% and a weighted F1 score of 0.90. The two classes of negative and positive sentiments contributed to the high performance, as the model appeared to leverage contextual and semantic information effectively. Nevertheless, performance on the neutral class, as measured by recall, remained very low at 0.13, reflecting class imbalance, despite the advances compared to classical models. The macro-averaged F1-score of 0.67 is an indication that predicting minority classes is still problematic even for transformer, based models. In general, DistilBERT is a better generalisation model than Logistic Regression and Random Forest.

Analysis and Discussion

The results showed that the ability to capture semantic context within textual data significantly impacted the performance of various sentiment classification models analysing Amazon product reviews. DistilBERT, with its transformer-based architecture, outperformed the other models. The accuracy and weighted F1 score obtained by DistilBERT are a testament to its strength in this aspect. This transformer model predicted sentiment more accurately. Logistic Regression operating on TF-IDF vectors provided a high standard baseline.

These findings align with the existing literature, which shows that transformer models outperform other approaches in sentiment analysis only when sufficient contextual information is available. The amount of overfitting in the results prompts us to reiterate that large-scale training sets and regularisation are necessary when working with these types of datasets. One major limitation of this analysis was the severe imbalance in the dataset, which severely limited neutral sentiment predictions.

As a rough guide, Logistic Regression is a lightweight, easily understood option that remains very capable, whereas DistilBERT is best suited to situations where accuracy is the top priority and computing power isn’t a problem.

Conclusion

We evaluated three different models (Logistic Regression, Random Forest, and DistilBERT) for Amazon product reviews by implementing the same pre-processing and evaluation metrics for all models to determine which model performed the best. Our analysis showed that while traditional Machine Learning techniques have a strong predictive power in most sentiment classes, they do not perform well on neutral reviews where class imbalance and TF-IDF features are barriers to performance. The strongest overall predictor was DistilBERT, which effectively captured both contextual and semantic information. Although Logistic Regression was the most simplistic of the three models, it proved to be an effective and affordable baseline model and had exceptional interpretability. This work has real-world applications in large-scale e-commerce customer feedback, product monitoring, and support systems. Future research should focus on creating better-balanced training datasets, investigating more advanced techniques for dealing with imbalanced classes, and expanding the research to include aspect-based sentiment analysis that is performed in multiple languages.

References

- Breiman, L. (2001) ‘Random Forests’, *Machine Learning*, 45(1), pp. 5–32. Available at: https://doi.org/10.1023/A:1010933404324

- Devlin, J., Chang, M.W., Lee, K. and Toutanova, K. (2019) *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Proceedings of NAACL-HLT 2019. Available at: https://arxiv.org/abs/1810.04805

- Kaggle (n.d.) *Amazon Reviews Dataset*. [Online] Available at: https://www.kaggle.com/

- McAuley, J. and Leskovec, J. (2015) *From Amateurs to Connoisseurs: Modeling the Evolution of User Expertise through Online Reviews*. Proceedings of the 24th International World Wide Web Conference (WWW). Available at: https://nijianmo.github.io/amazon/index.html

- Pang, B. and Lee, L. (2008) ‘Opinion Mining and Sentiment Analysis’, *Foundations and Trends in Information Retrieval*, 2(1–2), pp. 1–135. Available at: https://www.cs.cornell.edu/home/llee/omsa/omsa.pdf

- Sanh, V. et al. (2019) *DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter*. Available at: https://arxiv.org/abs/1910.01108

Appendix

Appendix A: Data Sources:

- Amazon Review Data from Kaggle

Appendix B: Data Preprocessing and Feature Engineering

- Removing Duplicates
- Removing NULL values
- Adding new features such as rating

Appendix C: Models

- Logistic Regression – 86% accuracy
- Random Forest – 85% accuracy
- DitilBERT - 90% accuracy