"Sentiment and Rating Prediction on Book Reviews: A Comparative Study of Classical and Boosting Models Using Numerical and Text Analytics"

Dataset1: Divesh sanjay patil
Msc in Data Analytics
National college of ireland, Dublin Ireland
x23401478

Dataset 2: Ruchita Rupesh Raut
Msc in Data Analytics
National college of ireland, Dublin Ireland
x24240702

Dataset 3:
Anuja Tawade
Msc in Data Analytics
National college of ireland, Dublin Ireland
x24257788

Abstract:
Platform such as Amazon, which provides the online book reviews gives valuable information and readers opinion, preferences and behaviours. Review both textual feedback and structure, meta data such as ratings, revival, sentiment, and verified purchase information. The main aim of this project is to analyse Amazon book review data to understand book, popularity and review behaviour and to examine whether machine learning models can effectively predict ratings using text and data features. Following the key methodology, which is knowledge discovering database, the data it was cleaned, transformed and reduced to meet the project requirements , where multiple machine learning modes were applied, including logistic regression, random forest, linear regression, XG boost, to perform multi classification of review ratings, models, performance was evaluated using accuracy, F1 score, confusion, matrix, and related metrics. To improve the transparency and trust in process, the shap explain ability technique were used to identify which textual and meta data features strongly powered prediction. This study demonstrates the importance of interpret with performance when applying machine learning to real world review data.

Keywords: Amazon Reviews, Text Mining, Machine Learning, Tf-IDF, Random Forest, Logistic Regression, Sentiment Analysis, SHAP, Multi-class Classification, Linear Regression , Lightgbm , XG boost, Log Transfer.

Introduction:

Dataset 1:

In today's world, where online book platforms like Amazon, which collect large amount of data through user ratings and reviews, making them an ideal source for analysing book popularity. in this project, I have analysed and English language, Amazon books data set to understand which features have powered popularity and whether the machine learning models can predict it correctly. I compared a simple machine learning model linear regression with the more advanced model, random forest and have used sharp to explain how the best model makes it prediction. So, the aim is not to only achieve gold prediction performance, but also clearly understand what factors drive the most for the book popularity. where my research question is 'using only book meta data can machine learning model predict books popularity (rating counts, text review, counts) and which meta data features are the strongest predictor of popularity?' Answer this question. To answer this question I followed the KDD methodology, which is known as knowledge discovery database. The result of the study highlights how combining book meta data with engaged best features, improve prediction performance and provides a clear understanding of book popularity and with the help of explain ability tool such as sharp, which helps tell that which factor influence the model prediction.

Dataset 2:

In day-to-day life, the products which we want to buy before deciding which product we have to buy the online review, play a major role in helping customers. The platform like Amazon, where book reviews contain rich information, together with text, star ratings, and meta data about the review and the review itself. For predicting this ratings using the machine learning model, which is an important challenge because it helps understand business about the customer behaviour which improve recommendation system as well as detect unusual or unreliable reviews .In this project, with the help of machine learning models, I have analyse Amazon book reviews which can automatically predict the star rating which is 125 on both text features and meter data features. Whereas the text features tells what the reviewer is saying .while meta data such as sentiment, review length ,title length ,verified purchase provides the additional context by combining this two types of information .I aim to build a stronger and more accurate predictive model My research question guides me for this work: "Can we accurately predict Amazon book review ratings using combine text best TFIDF features and meta data and which features contribute most to the models decision? "To Answer this question, I followed the KDD methodology, which is known as knowledge discovery in database. The interfere ability technique such as sharp, which is used to understand how different features hold the predictions. The results of this study tell and highlights how text and meta data together improve model accuracy and how well explain ability tools help to reveal the most important factor affecting the book rating prediction.

Dataset 3:

Online reviews are extremely important, acting as modern-day word-of-mouth that build trust, drive sales, influences buying decisions, with many rusting them more than personal recommendations or company marketing. Around 95% of consumers read reviews before purchasing. That is the main reason for our project, online reviews about books collected from platforms like Amazon. These reviews contain a lot of information about how readers interpret, enjoy and respond to the books they read. The language used in these reviews reflects personal touch, reading preferences, their habits, emotional state while writing this reviews and even behavioral patterns among different types of reviewers. The aim of this project is to understand how readers express their opinions and can we classify them based on text by using machine learning models and explore this immense information by asking the right questions like, "What can we learn from the patterns and language of the text reviews found in the Amazon book reviews, and how do these insights help us understand reader opinions, review helpfulness and review behavior?".

To answer these questions, the process begins with understanding the data, deciding the target variables, cleaning the data, transforming if necessary, and creating a clean text column to capture high-dimensional language features using TF-IDF. Since, after converting the text into TF-IDF features, the dataset becomes high-dimensional and sparse, so to handle this kind of data, we use the gradient boosting models that use decision trees to make predictions, which are XGBoost and LightGBM.

Related Work:

Dataset 1:

The recommended system for book review based on clustering algorithm uses the Amazon book reviews and applies Kimmins clustering method for grouping the review keywords which shows the result in a dashboard. It is useful for me because it supports the review ratings data to understand engagement and talking about the popularity prediction, it focuses more on clustering and keywords.[1]

book recommendation using machine learning methods based on Library loan records and Bilo graphic information. This uses ML models like random forest and SVM on booked as well as usage feature to build recommendation prediction as it uses the random florist for book related prediction task as this data set is library loans, but it is not Amazon review popularity.[2]

Amazon booking prediction and recommendation model uses the machine learning models such as classification and recommendation style methods. It is useful because it shows the model comparison idea and how Amazon book features can be used for prediction. The limitation is it often converts the problem

into rating classes which can lose the detailed compare to the regression style, popularity target.[3]
4) predicting online report purchases which compares multiple machine learning models on online purchase behaviour in an e-commerce setting as it supports doing modern comparison and using non-linear models for real world behaviour. Here the target is repeat purchase, not book popularity directly.[4]
5) book an introduction to statistical learning which gives me the linear regression baseline and tells why model works better. When patterns are non-linear. It is useful for my project as it shows a strong theory support for the choice of my linear regression versus random for evaluation.[5]

Dataset 2: This paper uses a large Amazon Kindle review data set which compares the logistic creation versus RNN and using the NLP pre-processing like cleaning, stop removal, limitation, et cetera. It suppose our choice of logistic regression as a strong baseline for the test problems as it is fast and easy to interpret. However, their meant task is of sentiment polarity, not full like one star 25 star Multiplex, so it is not directly solving our exact rating prediction objective. Still, it is useful for telling TFIDF plus logistic regression as a starting point.[6]
This work of the paper is close to the domain which is Amazon book and which uses TFIDF with logistic regression and random forest. It is very relevant because it discusses pre-processing plus feature, extraction and model comparison which highlights the big issue of class imbalance as balanced versus unbalanced data gives different information and the performance. Their focus is more on sentiment, labels and recommendation while target is multi star rating 125, but it strongly supports our decision to compare the logistic regression versus random forest and discuss the imbalance of the dataset.[7]
the studies applies sentiment analysis plus supervised machine learning model to detect the fake reviews and evaluate using metrics like accuracy, precision recall F1. It is useful because it shows how Amazon rivers are commonly model with classic machine learning pipelines. Hear their objective is fake review detection, not star prediction, so it supports methodology, evaluation choices, but not the prediction target.[8]
this paper directly aligned with the star rating idea which explains why predicting star ratings from brave text is difficult due to its bias and category revised. It also discusses the limitations of bag of words and proposes. Reach opinion-based recommendation. This supports our motivation for using both tech signals plus extra features, which also tells why performance can be limited. The limitations for this is it is more about alternative representation and research discussion, but it strengthens the why this is hard argument.[9]
this report describes the data set, pre-processing feature, engineering and multiple models, including machine, learning and deep learning plus evaluation for both binary and multi class settings. It is useful as it supports the idea of rating prediction which is commonly framed as classification problem and the model can be compare. So, the limitation here is it is more system or project oriented and less detailing control experiment, still, it helps to complete a pipeline approach such as data feature, modelling, and evaluation.[10]

Dataset 3:

Many researchers have studied sentiment analysis on book reviews, mainly to understand whether readers feel positively or negatively about the books they read. These researchers have explored a variety of basic machine learning models and simple text features to deep learning models and transformer-based models. Although this study shows progress, they also lack when it comes to handling large datasets and improving the model performances on real world noisy data.
Azhaguramyaa *et al.* used TF-IDF and Bag-of-Words with several classical classifiers such as SVM, Naïve Bayes, KNN, and Random Forest for book review sentiment prediction [11]. Their results showed that traditional classifiers can work well for small or medium sized datasets, but they struggle with understanding context and handling high-dimensional sparse data. This is where our project comes in, by applying XGBoost and LightGBM, which performs better large datasets and shows stronger accuracy than older models.
Singh *et al.* analyzed sentiment in over 800k Amazon reviews and used Logistic Regression and Random Forest after cleaning and handling the imbalance dataset [12]. The focus of their research was to remove the noise and balancing the classes. Unlike [12], our project includes SHAP, which helps understand which words influence the model's decisions.
Fredynand and Samosir used DistilBERT to detect sentiment toward different aspects of books [13]. Using transformer models helps understand the text much better but they also require powerful hardware to run. For this, our project takes a more practical approach by using TF-IDF with boosted tree models, which run faster and scale better in environments like Google Colab.
Fajar *et al.* used sentiment classification to improve book recommendations using collaborative filtering [14]. Their study shows how sentiment can support recommendations, but their models is simple and does not really provide detailed information. In our project, we have tried to build a simple recommendation system using the text.
Renukadevi et al. used Continuous Bag-of-Words approach with ML classifiers for book-review sentiment analysis [15]. Their models do not really scale well to large datasets, whereas in our project we used DASK to process the large data efficiently.
Across the studies reviewed, we came across some common issues, like most of them worked with small datasets, they relied more on traditional machine learning models, whereas we have learned from them and tried to use models which actually work better with high-dimensional text data and used metrics like Cohen's Kappa, Sensitivity, Specificity and many more to give a more accurate evaluation

KDD Methodology:

Dataset 1:
This project follows the KD methodology which is knowledge discovering database process. It gives the structured way to collect data clean it analyse it and finally create predictive models. The main aim of the methodology is to completely change Amazon book data into meaningful features and then train machine learning models to learn about the features for books popularity.
I. Data Cleaning:
The original data size is 11119 books containing the information such as rating review, speech, count, publication, details, and language quotes. To make sure about avoiding the messy data and consistency, only English language books were selected. non-English books is removed to avoid the inconsistency which would affect my project as well as it it can create the messy data. Books with the rating is zero where removed because the focus of the project is predicting popularity based on user interaction.

Fig 1 : Data Cleaning:

Dataset 2:
The original data size is of 72,7877 reviews, so after cleaning, filtering it and then pre-processing. Missing or incomplete rows, were removed. Balanced and high-quality of 10,000 reviews, data is selected to make about faster modelling.

The key variable selected for prediction is review text, which is mean signal for sentiment plus emotions, rating from one star to 5 star that is the target variable and verified purchase status, year, text length features such as word length, text length, title length. and the last sentiment polarity score from (-1 to +1) positive. This clean 10,000 record data set is formed the base for feature engineering and modelling.

```
[248]: #will load the file now and check for information of it
       dmn1=pd.read_csv(r"D:\DMML\review_10k.csv")  #reading the file
       dmn1.head(3)
```

[248]:

	rating	title	text	asin	parent_asin	user_id	helpful_vote	verified_purchase	date	year
0	5.0	Good	Just what I expected!	1542021472	1542021472	AFAHBYOMYBRSJNAYCR5P2PCUBMWQ	0	True	2023-09-13 22:45:55.547	2023
1	1.0	Marketed well but not a good book	Great idea and synopsis but not a good read. A...	1692354469	1692354469	AE72ELUQC3WVXSBO32T48ZEGWWLQ	0	False	2023-09-12 06:10:40.728	2023
2	4.0	Bent up books	Amazon typically has unbeatable prices on book...	059333583X	059333583X	AE5A3RKUXHJI2E3JU2HZZOOVZIQ	0	True	2023-09-08 03:22:58.541	2023

```
[246]: dmn1.shape
```

[246]: (10000, 10)

Fig 1: Data selection

Dataset 3:

The dataset used in this project was obtained from the Amazon Books Reviews created by Bekheet on Kaggle [6]. The dataset contains a large volume of real-world book reviews, including book titles, reviewer's names, dates when the review was posted, short summary of the actual review, unique IDs. This richness makes it suitable for analyzing sentiment, linguistic patterns and reviewer behavior, and for addressing the research questions defined in the study.

All columns from the dataset were kept except for the "Price" attribute, which was removed as it had no direct relevance to textual sentiment, helpfulness prediction or reviewer behavior analysis. The remaining attributes were essential for the sentiment analysis. We did create a new column named "Label", based on how the reviews were rated, 1(positively rated) or 0 (negatively rated), which enabled binary sentiment modelling with XGBoost and LightGBM. Also, the time column was in UNIX style format which was then converted to standard date-time format, which allowed comparisons of sentiment and review characteristics across different years.

Since, the dataset is too large, we used stratified sampling to ensure that the proportion of each class (positive or negative reviews) remains the same in the sample as in the entire dataset. The sample size provided a balanced compromise, ensuring the linguistic and behavioral patterns remained visible during modelling and evaluation.

	Missing Count	Missing %
Id	0	0.00
Title	208	0.01
Price	2518829	83.96
User_id	561787	18.73
profileName	561905	18.73
review/helpfulness	0	0.00
review/score	0	0.00
review/time	0	0.00
review/summary	407	0.01
review/text	8	0.00
review_date	0	0.00

Figure 1: Missing value data

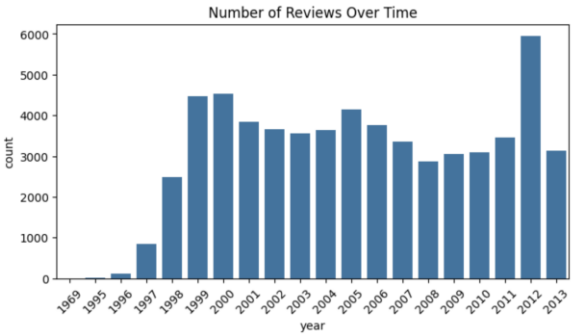


Figure 2: Number of reviews over time

II. Data preprocessing:

Dataset 1:

The data set was examined using info(), describe(), shape to understand data types .

During pre-processing the data, it was checked for missing values and duplicate and invalid entries,as my data set is numeric, so the key numeric variables such as rating counts and text review counts which shows the extreme skewness with a small number of books having very large values. To manage this issue log1p (logarithmic transformation) it was applied to reduce scheme and limit the power of outliers.Extra variable Books -age where take from publication dates to better capture time related effects.

Dataset 2:

Data was examined using info () and describe () to understand the data types, missing values and over a structure. Where 152 duplicates rows were identified and remove them with the help of drop_duplicate(). With the help of isna(). sum () the missing values were checked. Text cleaning includes the steps like conversion to lowercase, removing numbers, punctuation and special characters as well as stop words and the normalised white space to make the data prepared for TF-IDF. The target variable rating is converted into the integer format for classification. Review text is converted to string format with the help of str.replace(r's+', ' '), .str.strip() remove the unnecessary white space and special characters. The categorical values like true or false, it's converted into numeric label such as TRUE (1) FALSE (0), which allows machine learning model to use this for sentiment now this make sure that the text is ready for TF-IDF vectorisation.

Dataset 3:

Before moving to the modelling part, we must make sure that the dataset is structured to ensure consistency, usability and suitability for text-based sentiment modelling. For that, we checked for missing values, where we found that the "Price" column had 83% missing data, rendering it useless for meaningful analysis, the text field had 8 rows missing, so we dropped them safely with no loss of statistical integrity. Other attributes such as Title, Summary, Profile-name contained modest proportions of missing values, so instead of dropping them we decided to fill them with unknown title, unknown user or empty strings. This approach in turn maintained the completeness of the data.

That was all for the data part, we move on to the text cleaning part, which is a key part of our preprocessing. The review text was normalized to lowercase, HTML tokens were removes, URLs were stripped, unnecessary punctuations, extra whitespaces were cleaned. This reduced the noise and ensured that TF-IDF vectorization captured meaningful linguistics patterns. Finally, due to the large size of the dataset, a 2% stratified random sample was selected for modelling. This sampling maintained the distribution of sentiment classes while reducing the power, making dataset manageable for experimentation without compromising.

Overall, the cleaning process ensured that the dataset was complete, consistent and ready for high-dimensional text processing and gradient-boosting sentiment modelling.

III. Data Transformation (Feature Engineering):-

Dataset 1:

Features like reviews for rating Which catch the readers engagement behaviour .Rating per page, which tells the quality to book length , this features were created to increase predictive performance, and it represents the meaningful relationship rather than rely on raw data.So the final data set which consist of numeric variables, making it perfect for regression based machine learning models .The attribute publication date was converted to proper date time using to_datetime(). Log transformation is used for it which takes the large number of data, and then it is compressed into small range . as it is very useful, when data is very large and very small, which is just like the data set which I am using where there are many large values and many small for that purpose, the log transformation is used

A) Histogram:

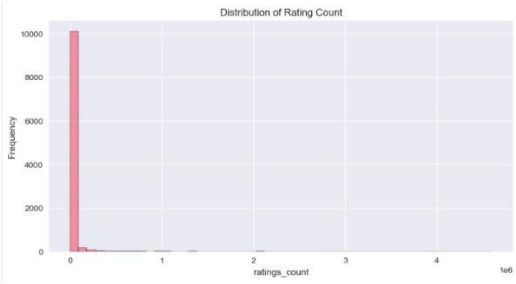


Fig 2:Distribution of Rating count

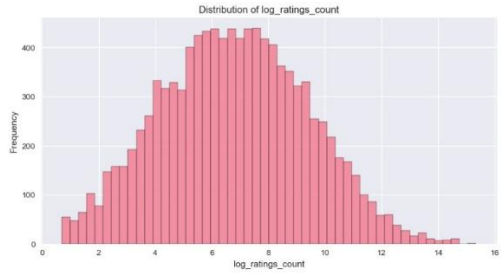


Fig 3:Distribution of log Rating count

The output of the histogram shows that one tall bar on the left and almost no bars on the right, which is extremely right skewed ,where it can be clearly seen a large number of books have low rating counts close to (0 to 500). Small number of books of extremely high rating counts. This tells he scheme Ness and outlaws which negatively affects the raw rating count, and it is not suitable for modelling due to extreme right skewness is to use log transformation.

The output of distribution of law rating counts shows the bell shape curve after applying lock transformation, which shows the even distribution. The values are been evenly distributed from 1 to 50 showing better variants, more stable patterns for machine learning models. After applying log transformation, the histogram looks balanced and symmetric, which is ideal for regression feature, importance, and interpretation.

c)Heatmap:

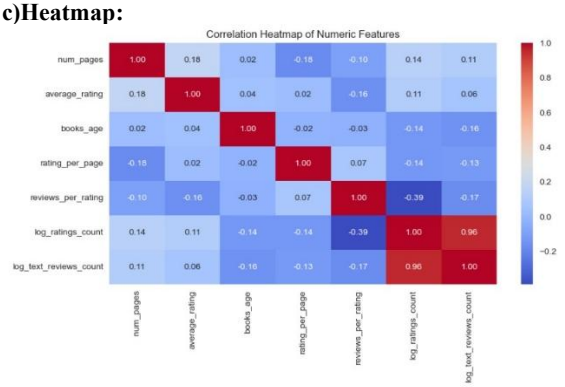


Fig 4: Distribution of Heat Map of Numeric features

The output of the heat map shows the correlation of two numeric values move together where the value ranges from plus one which is perfectly positive relationship which is red in colour and minus one which show the perfect negative relationship which is blue in colour and no relationship that is zero means very fake, which has been shown in white and light colour. The heat generally helps us to understand which feature are useful for prediction. The main target of feature is log rating count for predicting the popularity, so here in the output, we can see that the strongest correlation is 0.96 that is books with more written reviews, which also has more total ratings in the sense where we can say that reviews and ratings are almost perfect. Heat waves gives the understanding of data set as well as a feature selection like feature features to keep and which to remove. It also helps to predict a modern interpretation which is actually good for explanation and it helped me to choose the appropriate model.

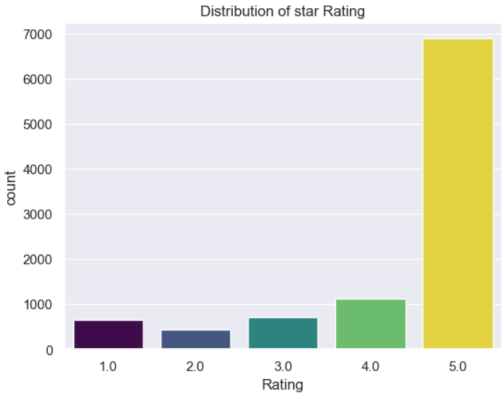
Dataset 2:

Feature engineering in my model was used to increase the performance and interpret ability. where the new features were created like text length feature which described the total characters in the review. The other feature is word length feature that is total number of words, title length feature which is length of review title, sentiment and verified purchase, which tells whether the reviver actually bought the book or not. This feature helps the model to learn about behavioural patterns over the text itself. VADER SentimentIntensityAnalyzer is applied to evaluate emotional tone of reviews. Compound sentiment score generated as a numeric feature telling positivity/negativity review language patterns. This features matter for stronger opinions; sentiment score provides the emotional context over star rating and numeric feature improve predictive performance of ML model.

After Adding the features, the multiple graphs are created for better understanding of the data.

Figure 2- shows the Rating Distribution which visualized the distribution of 1–5-star ratings. It identifies the heavily imbalanced classes and guides the model selection and balancing strategy.

Figure 3-word count analysis across star rating shows how review length changes with different star rating and how lower ratings like 1 star ,2 star have sorted reviews, which tells the negative feedback and higher rating with positive feedback.



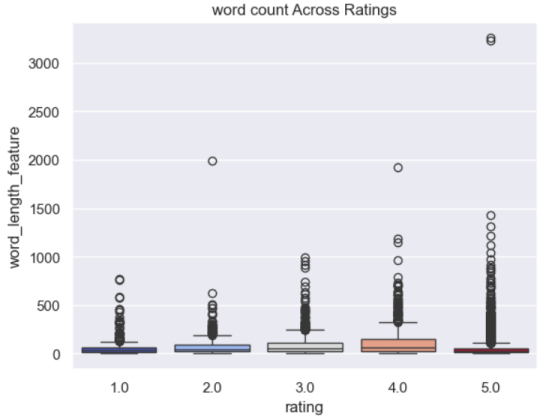


Fig 2: Distribution of star Rating

Fig 3: Word count Across Rating

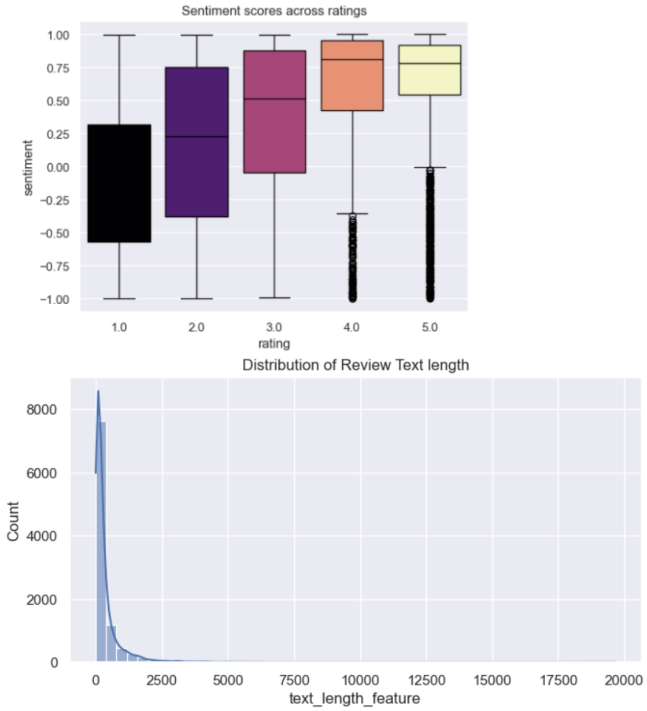


Fig 4: Sentiment scores across ratings

Fig 5: Distribution of Review text Length

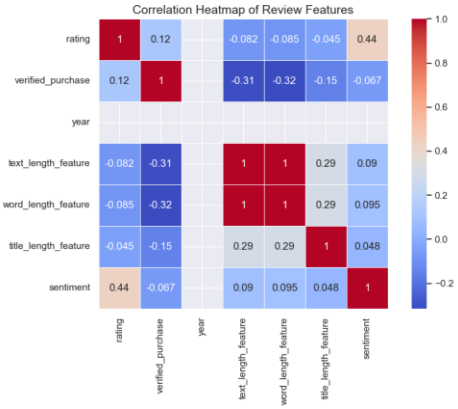


Fig 6: Correlation Heatmap of review features

Figure 4-outliers in the boxplot tells that a smaller number of users have long reviews instead of ratings, we can see that the it follows the upward trend. Sentiment score analysis across star rating shows the clear upward trend, which tells that sentiment is strong, productive feature for rating classification. One star shows the grouping in black colour as negative sentiment, 3-star show mixed and neutral, 4 and 5 star shows strong productive feature.

Figure 5- Histogram, which shows the distribution of review text length over the dataset. The heavy concentration below. Thousand character shows the reviews are short and long tail shows the very long creating a right-skewed distribution. This tells that users generally write short reviews. As well as highly detailed reviews are also there.

Figure 6-Heat map show how each feature related to target rating and other features. Where sentiment shows strongest positive correlation with rating (0.44), text length, word count, title length show correlation close to (0), verified purchase with rating (0.12) showing slightly more positive review from verified buyers.

Dataset 3:

Data Mining:

In the data mining stage, we focus on building the tree-based models which can identify sentiment patterns within book reviews.

After transforming the cleaned text using TF–IDF, two gradient-boosting models—XGBoost and LightGBM—were selected for sentiment classification. These algorithms were chosen because they are well-suited to handling the high-dimensional and sparse nature of TF–IDF matrices, and because they capture complex, non-linear relationships in language that simpler linear models often overlook. Additionally, both models offer interpretable feature importance scores, which support the project's objective of understanding which words contribute most strongly to positive or negative sentiment and exploring factors connected to review helpfulness.

In this stage, the main task was to learn predictive patterns from the review text that aligned with the project's research questions. Both XGBoost and LightGBM were trained on the TF–IDF representations of the review text using an 80/20 train–test split with fixed random seeds for reproducibility. XGBoost was applied directly on the sparse TF–IDF matrix, while LightGBM used a dense representation to support later explainability analysis. Model predictions were evaluated using accuracy, F1-score, confusion matrices, and Cohen's kappa, enabling a comparative understanding of how each algorithm performed on the sentiment classification task

Overall, the data mining stage allowed the project to uncover meaningful linguistics patterns embedded in the dataset, compare the strength of two tree-based gradient boosting models and generate insights that support the boarder analysis of reader sentiment, review helpfulness and reviewer behaviour.

IV. Model Preparation (Train-test Split):

Dataset 1:

The model preparation was used for the understanding of features where feature set X used the non-pages average rating books age, rating per page, revives per rating and feature, why which is the target variable is log rating count. The data split into 80% of training data and 20% of testing .Here the random state 42, which is used and it makes sure about split is reproducible for consistent results.

	num_pages	average_rating	books_age	rating_per_page	reviews_per_rating
0	652	4.57	19	0.006998	0.013166
1	870	4.49	21	0.005155	0.013571
2	352	4.42	22	0.012521	0.038522
3	435	4.56	21	0.010459	0.015526
4	2690	4.78	21	0.001776	0.003959

```
#split data into train and test

x_train,x_test,y_train,y_test=train_test_split(
    X,y,
    test_size=0.2,
    train_size=0.8,                    #training data tech the ML model
    random_state=42                    #testing data check how the model learn.
                                       #0.2(20%)=test, 0.8(80%)=training , random_state it make sure the split is reproducible.
)

x_train.shape,x_test.shape
```

```
((8369, 5), (2093, 5))
```

Fig 5: X test Variables

Dataset 2:

Feature set (X) used to review text verified purchase year length with features, sentiment score, whereas feature (Y) is the target variable for prediction and For the train test split the data is split into 80% of training data and 20% of testing data .While training while training, the data stratify = Y is used to make sure that rating distribution remain consistent in both training and test sets and random state = 42 make sure about split is reproducible for consistent results.

V. Modelling:

Dataset 1:

For this project Machine Learning models were trained and compared this is Linear Regression and Random Forest.

A) Linear Regression:

Linear regression is a basic machine learning model which draws a straight line through the data to make predictions in the project.

Linear regression tries to understand how features like average rating number of pages of book age affect book popularity as a linear regression cannot fully get these patterns, that is why it's accuracy, lower. It tries to learn how much each feature increases or decreases the popularity score. for example, for every extra page, popularity changes by small amount or higher rating increases popularity in a straight-line way, but on Amazon book popularity, this do not follow such simple patterns engagement behaviour is complex because of this the linear regression just tells small part of the variation in book popularity, and it gives the low accuracy. The linear regression model shows the performance such as RMSE (root mean square error)2.3139 MAE (mean absolute error) which is 1.8083 and R Square gives 0.2086.

```
# now apply first model Linear Regression

linear_regression_model=LinearRegression()            #here creating linear regression model
linear_regression_model.fit(x_train,y_train)          #through'.fit' get to know tech the model which kind of books are popular based on traini
y_prediction_lr=linear_regression_model.predict(x_test) #make the prediction on unseen data means on test data
print_regression_metrics(y_test,y_prediction_lr,'Linear Regression')  #here prints calculated regression metrics for linear regression
```

```
Linear Regression performance:
RMSE:2.3139
MAE:1.8083
R^2:0.2086
----------------------------------
```

Fig 6: Linear Regression

B) Random Forest:

After getting the accuracy low, the next model applied was Random forest, which is the most powerful model which works by building many decision, trees and combining their results. It makes better handling with non-linear and Messy real world data like predicting popularity.

So in the project, random forest performed better because it understood interaction between features such as how reviews per rating and average rating together, Powered popularity .

The results of random forest regression performance is RMSE (Root mean Square error) 1.0956 , MAE (mean absolute error) that is 1.4930 and R2 that is 0.4633.

```
#now apply second model RandomForest Regressor

randomforest_model=RandomForestRegressor(       # creating the random forest model
    n_estimators=200,                           #number of trees build the 200 trees in the forest, here 200 is a good balance for this dataset
    max_depth=None,                           # allow the model learn complex and depth patterns
    random_state=42,                          #set_random state here
    n_jobs=-1                                 #it's just increase speed ,do not change the accuracy.
)

randomforest_model.fit(x_train,y_train)     #each and every tree gets a random sample of rows through x_train
y_prediction_rf=randomforest_model.predict(x_test)  #make the prediction on unseen data means on test data
print_regression_metrics(y_test,y_prediction_rf, 'Random Forest Regressor' )  #here prints calculated regression metrics for random forest.
```

```
Random Forest Regressor performance:
RMSE:1.9056
MAE:1.4930
R^2:0.4633
----------------------------------
```

Fig 7: Random Forest

C) Model Comparison of Random Forest and Linear Regression:

	model1	RMSE	MAE	R2
0	Linear Regression	2.313915	1.808256	0.208586
1	Random Forest	1.905587	1.493010	0.463257

Fig 8 : Model Comparison Table

The comparison of linear regression and random forest show that value of linear regression RMSE is 2.313915. Which tells that it makes an average prediction error of + -2.31 low rating units, which is quite larger. whereas MAE 1.808 tells that on average predictions are 1.8 units away from the truth value. R square which is equal to 0.2085 that means the model only explain 20% of the variation in the popularity.Where by comparing the three values linear regression is weak and it cannot capture the pattern in the data. It performs poorly because the features are non-linear relationship. The second model random forest where RMSE is 1.095587 that is much smaller error compared to linear regression where MAE equals to 1.49, which means that on average, the prediction error drop by -18% whereas R square which is 0.463 explain 46% of the variation in book popularity. This is a huge improvement over the linear regression by 20%..Comparing both of the model, the forest is the best model because it build many trees and learn complex patterns like branches, whereas it has predicted the book popularity more accurately as compare to linear regression as a random forest were best with the countless data . It handles the feature interaction and also it handles outliers and skewed data.

D) Random Forest Feature Importance.

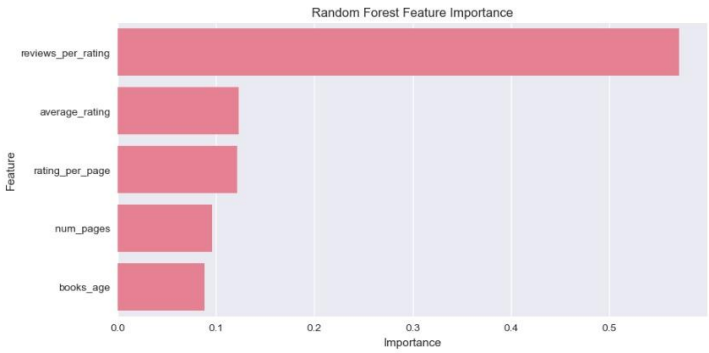


Fig 9: Random Forest Feature Importance

The output of feature importance shows that which features were most useful for predicting the target variable. The bar chart show the result like Reviews_per_rating. It is the most important feature which alone contributes 57% of the prediction Power that basically means if abook get many ratings, but very fewer written reviews that are very popular and if reviews_per_rating is high, books are less popular, so this is important because this ratio captures reader engagement behaviour and this is the strongest signal for predicting popularity. Whereas the average_rating shows the books with some of the higher average rating, which tends to be more talked about and get more total ratings,but we always have to remember that popularity does not always equals quality that's why it's important, but not the strongest as a compared to reviews_per_rating and talking about rating_per_page. It tells that how effectively a book clicks rating compare to it. Less shorter books might get more rating faster, which makes a meaningful feature. Talking about the num_pages, where there is a small influence of page count, we can say that very long books may get fewer readers, whereas the books with less pages are more accessible, but whereas we can say that page count doesn't actually determine the popularity. Books which means older books is equal to few new ratings, but again popularity is not been based on the age. As example, Harry Potter is old, but it is still very popular, so we can say that it is the least important feature. The use of random forest feature importance is used because it helped to understand which feature matters the most and it's supports the performance better. It helps with the shap values letter. When we actually you know, use the shap for it, which is good for a model explanation and it also let us know that which features are important and which features I need to remove.

Dataset 2:

For this project to machine, learning models were trained and compared that is logistic regression (balanced) and random forest.

A) Logistic regression (Balanced): -

This model handles high dimensional takes data very well. Logistic regression works well when using TF-IDF, which is text data balance minority classes using class_weight='balanced' and also it gives clear interpretability via coefficients. The model logistic regression gives the result as accuracy 0.517 and macro F1 -score 0.271 where macro F1 shows the modern struggles with smaller classes. By achieving precision, 0.79, recall 0.65 ,F1-score 0.71 model perform well for five star. Performance is big for 1 to 4 Star lowering rating. Weight average remains higher because five reviews dominate the data set. The confusion matrix most

predictions are concentrated in 5-star column showing the strong bias towards majority classes and this tell us use more powerful model which can handle interaction and feature complexity better.

```
===Logistic Regression (Balanced)===
Accuracy: 0.517
Macro F1: 0.271

Classification Report :
```

	precision	recall	f1-score	support
1.0	0.13	0.30	0.18	131
2.0	0.10	0.07	0.08	89
3.0	0.14	0.28	0.18	141
4.0	0.22	0.18	0.20	223
5.0	0.79	0.65	0.71	1378
accuracy			0.52	1962
macro avg	0.28	0.29	0.27	1962
weighted avg	0.61	0.52	0.55	1962

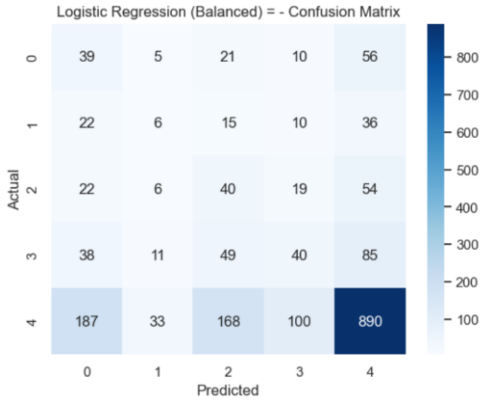


Fig 7:

Logistic regression Results

B) Random Forest:

This model is used after the logistic regression as logistic regression was not able to show the proper accuracy, so to handle the interaction and the complex pattern, the random forest is used by model which captures the non-linear patterns so it works well with the meter. Data features provide feature importance and it uses only numeric features not raw text because random for does not work well with high dimensional, TF-IDF matrices and also it goes with logistic regression by offering a tree -based perspective.

```
===Random Forest===
Accuracy: 0.681
Macro F1: 0.28

Classification Report :
```

	precision	recall	f1-score	support
1.0	0.29	0.28	0.28	131
2.0	0.09	0.03	0.05	89
3.0	0.18	0.06	0.09	141
4.0	0.26	0.09	0.14	223
5.0	0.76	0.92	0.83	1378
accuracy			0.68	1962
macro avg	0.32	0.28	0.28	1962
weighted avg	0.60	0.68	0.63	1962

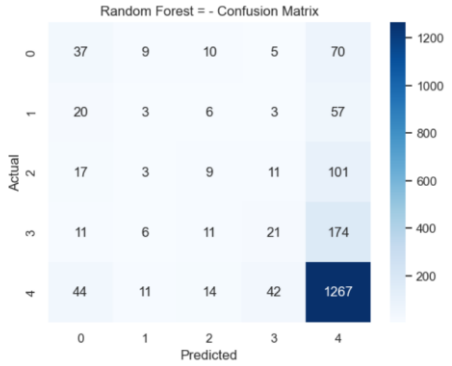


Fig 8: Random Forest result

Accuracy with 0.681, macro F1-score with 0.28. This both performs better than logistic regression, mainly capturing non-linear patterns and interaction. The small amount of improvement over logistic regression of 1 to 4 star through performance of smaller classes. It predicts the 5-star review strongly for precision 0.76, recall 0.92, F1 Score 0.83. Random forest is better than large derogation with higher accuracy and better handling of complex pattern.

C) Model Comparison of Random Forest and Logistic Regression balanced:

Logistic regression is used for its effectiveness and interpret with the text data while random forest is used to capture non-linear patterns in meta data that allows sharp base, explainability.Logistic Regression (Balanced) fast works well with linear patterns and interpretable. Logistic Regression fails to distinguish lower rating while random forest shows the small improvement in that. Random Forest handles the mixed features better, captures non-linear relationship with higher accuracy. Random Forest is better model as it delivers higher accuracy, stronger F1 performance and handles non-linear patter and as well as features interaction more effectively.

	Model	Accuracy	Macro F1	Preci
0	Logistic Regression (Balanced)	0.517	0.271	
1	Random Forest	0.681	0.280	

Fig 9: Model Result Comparisons

Dataset 3:

XGBoost Model:

Evaluation:

The XGBoost model performed well overall, achieving an accuracy of 0.855. the F1-score of 0.915 shows that it handled the positive class effectively. Cohen's Kappa of 0.433 indicates moderate agreement beyond chance, which is expected because the dataset is imbalanced.

```
===== XGBoost =====
Accuracy: 0.8553833333333334
F1: 0.9153636815871871
Kappa: 0.432995265990762
Sensitivity: 0.979010181939576
Specificity: 0.36456262425447317
Confusion:
 [[ 4401  7671]
 [ 1006 46922]]
```

Figure 4: XGBoost output

From the confusion matrix, the model correctly identified 46,922 positive reviews, giving a very high sensitivity of 0.979. this means XGBoost is extremely good at recognising positive sentiment. However, it struggled more with negative reviews. It correctly predicted 4401 negative reviews but misclassified 7671, giving a lower specificity of 0.365. this shows that negative sentiment was harder for the model to detect-something very common in review datasets where positive reviews are more frequent and clearer in wording.

Overall, XGBoost performs strongly for positive sentiment but finds subtle or mixed negative reviews more challenging.

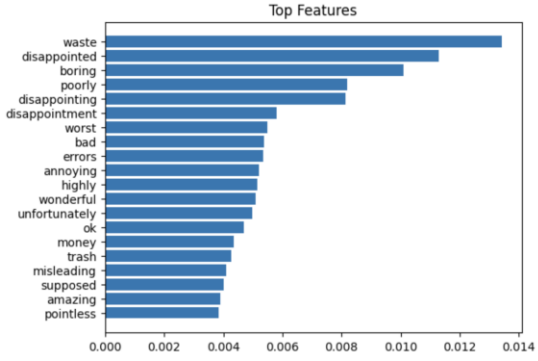


Figure 5: XGBoost features

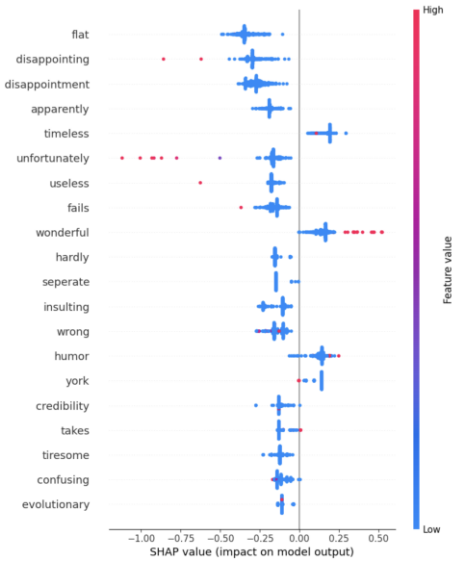


Figure 6: SHAP for XGBoost

SHAP was used to understand why XGBoost made certain predictions. The results showed that strong negative words like “waste,” “disappointed,” “boring,” and “worst” had the biggest impact on predicting negative sentiment. These clear emotional terms helped the model easily recognise negative reviews. Positive reviews, on the other hand, used more varied wording. Only a few positive words—such as “wonderful,” “highly,” and “amazing”—appeared as important features. This explains why the model sometimes struggled with subtle or mixed positive language.

Overall, SHAP helped show which words influenced the model’s decisions and confirmed that XGBoost relies mostly on strong, direct sentiment words.

LightGBM Model:

Evaluation:

The LightGBM model performed very well, with an accuracy of 0.871, an F1-score of 0.923, and a Cohen’s kappa of 0.536, being stronger compared to other models. LightGBM was especially good at detecting positive sentiment, achieving a high recall of 0.97.Like many sentiment models, LightGBM found negative reviews more difficult. It correctly identified 1,191 negative reviews but misclassified 1,261, giving a recall of 0.49 for the negative class. This is a common issue because positive reviews are more frequent and clearer, while negative reviews are often subtle. Even so, the F1-score of 0.77 shows that the model still performs reasonably well across both classes.

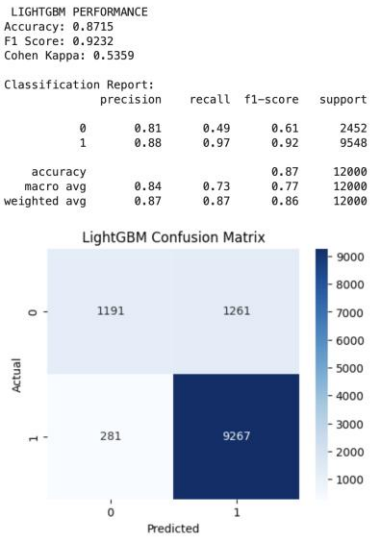


Figure 7: LightGBM output

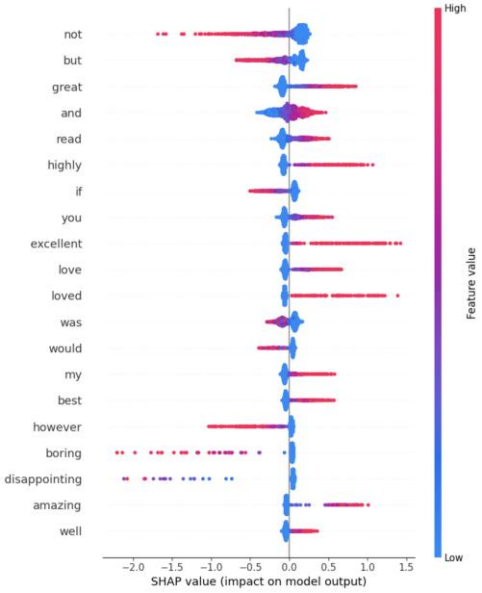


Figure 8: SHAP for LightGBM

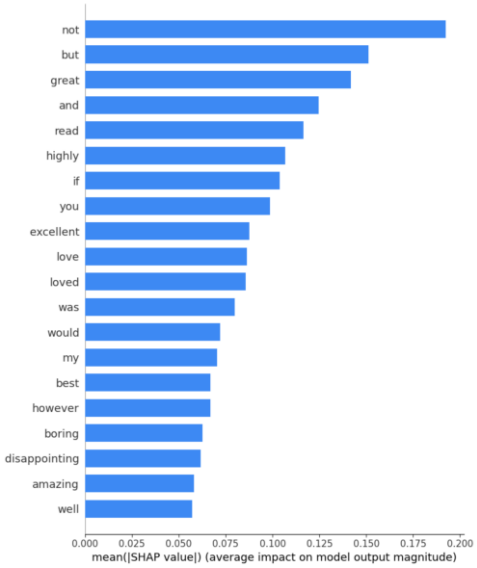


Figure 9: LightGBM features

VI. Interpretability(SHAP):

Dataset 1:

Shap (SHapley Additive exPlanations) explains how random forest model decides book popularity by showing the effect of each feature on the prediction. As we got the better model as random forest, the sharp is been applied on the random first.

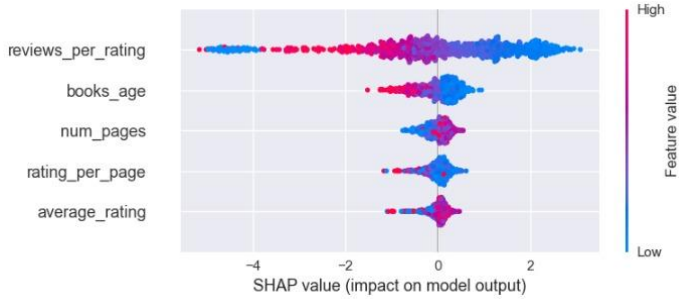


Fig 10: Shap Interpretability

The output shows that the reviews per rating is the most important feature and has widest spread, so it involves predictions the most. High values which is red in colour, push prediction strongly to the right and the low values which is blue in colour push predictions to the left which is lower popularity. This means that books with more revive Spitting are predicted to be much more popular. Book age, where blue colour mostly reduce popularity and red colour slightly increase popularity, this shows that the recent books tend to attract more attention .Num pages, where we can see that medium length books are neutral, very long or very short books can lowerly affect popularity .Rating per page shows that higher values slightly increases popularity, and the involvement is smaller as compare to the top features .Average rating shows that it has a smallest impact overall, even high rating do not guarantee high popularity. So this confirms that rating alone is not enough to make the book popular. It shows that user engagement which is review for rating. It's more important that average rating and also the random forest has successfully captured this complex patterns, making it both accurate and explainable.

A) Shap Attributes features:

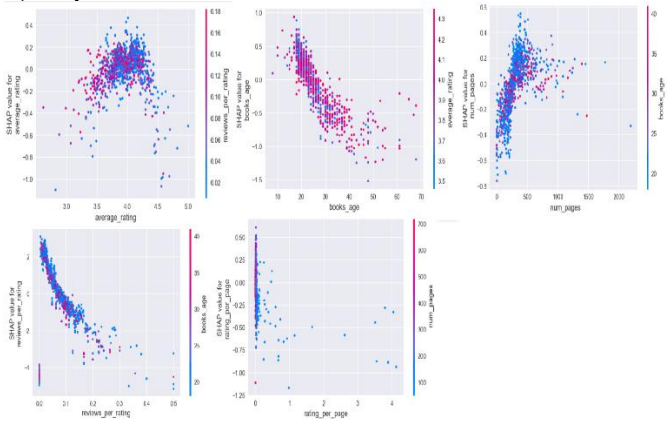


Fig 11 : Features

Here, in this output sharp dependency plots shows feature by feature. So first feature is average rating. It shows how rising book popularity but it's a reading is a not sufficient. It's not oh enough. Books with almost similar ratings can still very different popularity levels. Second feature is showing books age where older books show to reduce popularity, whereas new books usually contribute positive popularity. Third feature, which is a number of pages, shows the medium length book. Slowly increases popularity. But very long books do not show any extra benefits. Review per rating, which is the most important, were. It shows that books with higher assurance where we can say more reviews per rating strongly increases popularity. Which basically shows that rating alone don't matter more as compared to reader Interaction matters more. Rating per page has a small and different effect. That means quality, which is related to length showing a small role.

Dataset 2:

Shap (sHapley Additive exPlanations) shows how each feature contributes to a model's prediction**.** Shap is used on Random Forest Classifier as it works well with tree models as it can calculate exact feature contribution, give more interpretation and explain non-linear relationships. Shap summary plot shows sentiment has strongest positive influence on predicting higher rating (red colour are far right). Red and pink colour with High feature value and blue colour with Low feature value. Longer reviews tends to push predictions towards higher ratings. Verified purchase influences ratings but less strongly and the Year with no impact.

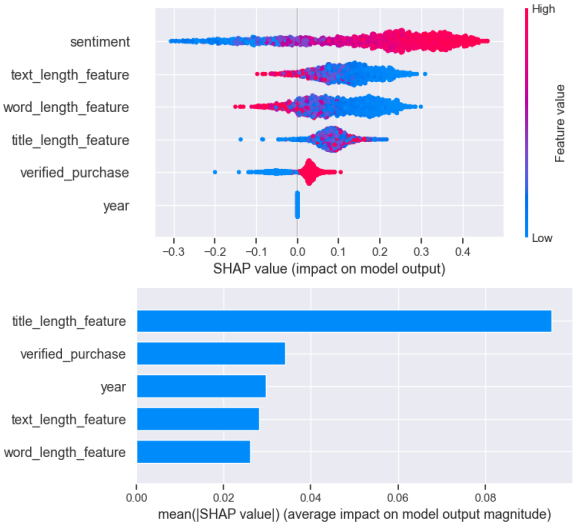


Fig 10: Shap interpretability

Fig 11: Shap bar plot

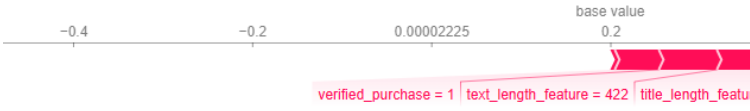


Fig 11. Shap Force plot

Shap bar plot and force plot:

The Global bar plots which highlight the overall feature importance in the model. The Global bar plot which gives average importance over all classes where title length feature has the most average impact then comes the verified purchase, year, text length feature and word length. The Shap force plot which is local explanation for one review it explains the one prediction where rating probability is 0.68 for that some features like high sentiment score, long text length, verified purchase and long title pushed the prediction up which is shown in RED. This makes model transparent. And shows what factors influenced each rating prediction.

Global Feature Importance (logistic Regression):

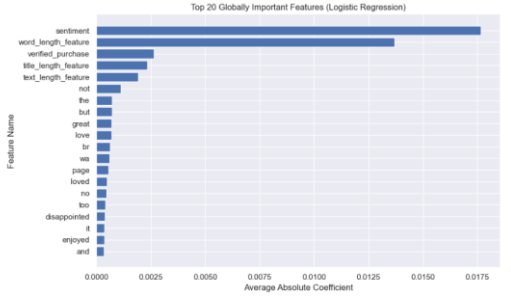


Fig 12. Top 20 Global feature Importance

The output of the graph shows the overall most important features used by the logistic regression model to predict Amazon book review ratings. Where the sentiment is the most important features with the positive sentiment, they are more likely to receive higher star ratings while negative sentiments lead to lower star ratings. Review length features are highly impact long and more detail reviews gives the stronger signal for predicting ratings where verified purchase has moderate impact and individual words such as great love, enjoyed disappoint. They appear, but they have less power than overall sentiment and structure.

Dataset 3:

SHAP analysis helped explain how LightGBM makes predictions. Important words included a mix of positive terms ("great," "excellent," "love," "amazing," etc.) and key connectors or negations such as "not" and "but." The word "not" had the strongest influence, as it can flip the meaning of otherwise positive words (e.g., "not great").LightGBM captured subtle positive expressions better than XGBoost, likely due to how it grows its trees. It also responded strongly to contextual words such as "but" and "and," suggesting that LightGBM pays attention to the structure of sentences, not only individual keywords. Negative words like "boring" and "disappointing" did appear, but they had lower influence overall the analysis, which helps explain why the model sometimes confused mild criticism with positive sentiment. Overall, LightGBM provides useful insights through SHAP, showing sentiment words and language patterns that influence predictions.

Model Comparison:

Aspect	XGBoost	LightGBM
Overall Performance	Good performance; especially strong at predicting positive reviews	Higher accuracy, F1-score, and kappa; stronger overall model
Positive Class (Sensitivity/Recall)	Very high recall (excellent at detecting positive sentiment)	Also, strong recall, performs consistently well
Negative Class Performance	Weaker; struggles with subtle or mixed negative sentiment	Better than XGBoost but still challenged by subtle negativity
Key SHAP Features	Relies heavily on **explicit negative words**: "waste," "disappointed," "boring"	Captures **contextual cues** and interactions: "not," "but," "great," "highly"
Interpretability	Clear patterns based on strong polarity words	More nuanced, captures structure and context within sentences
Strengths	Excellent at identifying strong, direct sentiment (especially negative words)	Better at modelling subtle sentiment cues and language variation
Weaknesses	Struggles with indirect or soft negative reviews	Sometimes misinterprets mild criticism as positive
Overall Insight	More sensitive to obvious sentiment signals	More flexible and context-aware; better general agreement with true labels

Conclusion:

Dataset 1:

This project shows that Amazon book popularity can be predicted using metadata features, with reviews_per_rating being the most important factor. Books that receive more written reviews relative to ratings tend to be significantly more popular. Other features like average rating, book age, and number of pages have smaller effects. Between the two models tested, Random Forest performed much better than Linear Regression, proving that book popularity follows a non-linear pattern. SHAP analysis confirmed these findings by clearly explaining how each feature influences predictions. Overall, the study concludes that reader engagement drives popularity, and Random Forest is the most suitable model for predicting book popularity in this dataset.

Dataset 2:

The conclusion for over all project shows that the results shows that Amazon book review ratings can be predicted with average accuracy of 68% with the best model Random Forest.
Where Sentiment score, review text length, word count, title length and weather the reviewer is verified purchase these features contributes most towards prediction. SHAP Analysis confirms that sentiment is the strongest with longer and more review which pushing the predictions towards the higher star ratings.

Dataset 3:

This study explored how sentiment appears in Amazon book reviews using TI-IDF and two boosting models, XGBoost and LightGBM. Both models performed well, but LightGBM was slightly stronger overall, while XGBoost was especially good at detecting positive reviews. SHAP analysis showed that certain words had a strong influence on predictions-negative terms like "waste", "boring" and "disappointed" pushed reviews toward negative sentiment, while positive words like "great", "excellent" and "love" pushed them toward positive sentiment. These results help us understand how readers express their opinions and show the value of using interpretable machine-learning methods on large review datasets.

Future Work:
Dataset 1:

The future work is built up on the findings of the study by exploring the additional data resources, modelling technique and interpret methods where adding text feature from reviews(TF-IDF sentiment scores) which would combine them with meta data to improve the popularity prediction. The stronger models like XG boost /lightGBM and tune hyper parameters using cross validation to boost performance. And also using of features like publication, year decade trends, which would test whether the popularity drivers change across different time, periods and using the sharp for comparing with another method to improve the interpretability.

Dataset 2:

As the results of the project are encouraging, but still several additions would be inspect in future work. firstly, technique such as SMOTE or other relying methods could be applied for handling class imbalance as well as to enhance the prediction of minority rating classes from (1 star to 3 stars). Secondly, the more advanced text representations models like BERT that could be used for deeper semantic meaning for TF-IDF features. The further improvements may also be achieved through systematic hyper parameter optimisation and addition of rich meta data features like behaviour or readability score.These addition could lead to more correct predictions variable and deeper information into customer review behaviour.

Dataset 3:

Future work for our project could explore transformer-based models such as BERT to capture deeper textual context and improve the finding of negative sentiment. Addressing class imbalance and extending the analysis to reviewer behaviour, or helpfulness prediction would further enhance model robustness. Embedding-based similarity methods could also support developing a more advanced book recommendation system.

References:

Dataset 1:

[1]Works CitedAlam, Md. Zehan, and Tonmoy Roy. "Predicting Online Repeat Purchases: A Comparative Analysis of Machine Learning Algorithms." 2025 International Conference on Electrical, Computer and Communication Engineering (ECCE), 13 Feb. 2025, pp. 1–6, https://doi.org/10.1109/ecce64574.2025.11013423. Accessed 12 Dec. 2025.

[2]Lin, Hsiu-Ping, et al. Amazon Books Rating Prediction & Recommendation Model.

[3]Tsuji, Keita, et al. "Book Recommendation Using Machine Learning Methods Based on Library Loan Records and Bibliographic Information." 2014 IIAI 3rd International Conference on Advanced Applied Informatics, Aug. 2014, pp. 76–79, https://doi.org/10.1109/iiai-aai.2014.26. Accessed 12 Dec. 2025.

[4]Udariansyah, Devi. "Recommender System for Book Review Based on Clustering Algorithms." Journal of Applied Data Sciences, vol. 6, no. 1, 1 Jan. 2024, pp. 225–235, https://doi.org/10.47738/jads.v6i1.492. Accessed 12 Dec. 2025.

[5]James, Gareth, et al. An Introduction to Statistical Learning. Springer, 8 Sept. 2023.

Dataset 2:

[6]Works CitedAkshara, Sikha, et al. "A Small Comparative Study of Machine Learning Algorithms in the Detection of Fake Reviews of Amazon Products." 2023 6th International Conference on Contemporary Computing and Informatics (IC3I), 14 Sept. 2023, pp. 2258–2263, https://doi.org/10.1109/ic3i59117.2023.10398096. Accessed 12 Dec. 2025.

[7]P, Devaki, et al. "Sentiment Analysis and Recommendation of Book Reviews." 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT), 7 Oct. 2022, pp. 1–6, https://doi.org/10.1109/gcat55367.2022.9971996. Accessed 12 Dec. 2025.

[7]Singh, Ashish Kumar, et al. "Amazon Kindle Review Sentiment Analysis." 2024 IEEE Recent Advances in Intelligent Computational Systems (RAICS), 16 May 2024, pp. 1–6, https://doi.org/10.1109/raics61201.2024.10689900. Accessed 12 Dec. 2025.

[8]Xing, Fan, et al. "Social Media Text Sentiment Analysis: Exploration of Machine Learning Methods." 2024 IEEE 1st Karachi Section Humanitarian Technology Conference (KHI-HTC), 8 Jan. 2024, pp. 1–6, https://doi.org/10.1109/khi-htc60760.2024.10481912. Accessed 12 Dec. 2025.

[9]T. Laksanai, J. Puiplung, C. Chanat, D. Singhchalern, T. Udtha, A. Chuenjitwanich, W. Wuttisittikoon, and R. Lamlompol, "A Comparative Analysis of Machine Learning Models for Domain Adaptation in Multiclass Sentiment Classification,"

[10]ECTI Transactions on Computer and Information Technology, vol. 17, no. 2, pp. 184–190, Apr. 2023.

Dataset 3:

[1] V. R. Azhaguramyaa *et al.*, "Sentiment Analysis on Book Reviews Using Machine Learning Techniques," *ICACCS*, 2022.

[12] A. K. Singh *et al.*, "Amazon Kindle Review Sentiment Analysis," *IEEE RAICS*, 2024.

[13] D. Ferdynand and S. Samosir, "Aspect-Based Sentiment Analysis on Amazon Book Review Using DistilBERT," 2023.

[14] M. Fajar *et al.*, "Optimizing Book Recommendation Systems through Integration of Sentiment Analysis," 2023.

[15] R. Renukadevi *et al.*, "Sentimental Analysis with Continuous Bag-of-Words for Book Reviews," 2021.

[16] M. Bakhet, "Amazon Books Reviews," *Kaggle*, 2019. [Online] https://www.kaggle.com/datasets/mohamedbakhet/amazon-books-reviews

Summary Contribution:

Member 1 :(Divesh Patil)
Student Id: x23401478

My project analysed Amazon books meta dataset through Kaggle to understand, which factor influencing the most for popularity as well as to predict the popularity with the help machine learning methods in my project, I have used the KDD methodology for make sure a structure approach to data selection, modelling, pre-processing and interpretation. I have used two models in my project like linear regression and random Forest integration is used for baseline model while random forest capturing non-linear patterns in the data so random forest showing better performance as compare to the linear regression because it captures the non-linear patterns specially reviews per rating, where found the very essential factor through SHAP based explain. I have found the relevant research paper using scopus and all clean data code, outputs, references, and PPT where organise and share via Google Drive link as well as the project deliverables include a written report and presentation. PPT summarising the main features.

Member 2:(Ruchita Raut)
Student Id: x24240702

For my project, I used Amazon book review data which was to predict the review ratings and understanding the key factors. Where Relevant Literature papers was taken from the Scopus. The KDD methodology was followed, for cleaning, feature extraction, modelling, evaluation and interpretation, Machine learning models including Logistic Regression (Balanced) and Random Forest Were implemented and compared and got the random forest as the best model. Using Shap the model explainability which tells the clear information how text and metadata feature powered predictions. Which tells me which were the main and the Project outcomes were presented during a live presentation, and a structured technical report were created where all code, results and visualizations were shared via a Drive link for transparency.

Member 3:(Anuja Tawde)
Student Id: x24257788

For this project, I gathered the dataset from Kaggle, and I was tasked with the text analytics part of the project. I followed the KDD methodology for the pre-processing part, analysis and interpretation of the data. I also looked up research papers related

to text and sentiment analysis of book reviews on IEEE Xplore and Scopus, to understand the background and methodology sections. Along the way, I put together a PowerPoint presentation to clearly summarise the process and key findings. Also had to write a report based on our findings to clearly mention the results.