

# Amazon Product Reviews: Predicting Rating Scores with Binary and Multiclass Classifiers and Clustering Categories with Reviews

Wenxin Zhao (Rose)  
Dartmouth College  
14 North Main Street Hanover, New Hampshire, USA  
wenxin.zhao.gr@dartmouth.edu

## Abstract

*A popular application in computational linguistics research is sentiment analysis of product reviews. In this paper, we apply machine learning models to explore if the Amazon product reviews can indicate the rating of the products given by the customers, and suggest the product categories. We apply Logistic Regression, Support Vector Machines, Random Forest models as both binary and multiclass classifiers, and K-Means, DBSCAN, and Hierarchical Clustering for clustering. We acquired the highest Macro F1 score with Linear Regression, of 0.74, 0.98, 0.83, 0.77, respectively for binary classification of each rating cutoff, and 0.53 for a multiclass classification. DBSCAN yielded the highest Silhouette score of 0.99.*

## 1. Introduction

Nowadays, online shopping has become inseparable in our daily lives. Product reviews provide valuable feedback for both consumers and sellers. However, the volume of product reviews on major e-commerce platforms like Amazon has exploded. Therefore, businesses need effective automated methods for the analysis of text data. In this paper, we explore the use of machine learning algorithms to classify Amazon product review ratings and clustering review types. Our goal is to develop models that can accurately predict the rating given by a user and product category based on their review.

To accomplish this, we use a dataset of Amazon product reviews, and train several machine learning models, including logistic regression, support vector machines, and random forest. We then evaluate the performance of these models using various metrics including Macro F1, accuracy, confusion matrix, ROC and AUC. Our results demonstrate the effectiveness of machine learning in classifying product review ratings and provide insights into the most effective approaches for this task.

## 2. Related Work

There are many previous work done on this topic. Gope in the paper [1] used a dataset of 34,000 Amazon reviews, explored both classical ML algorithms and deep learning frameworks, and concluded that Random Forest classifier was the best performing one among classical algorithms. In another similar attempt, Haque in the paper [2] concluded that SVM performed the best. Both papers used Tfidf Vectorizer on review text data and applied various feature-extraction techniques such as removing stop words, cleaning text, stemming and so on. However, both paper approached the classification as binary, positive for rating 1 and 2, negative for rating 4 and 5, and discard the neutral rating of 3.

In this paper, we provide more holistic classifications that include all four kinds of rating cutoff as well as predicting multiclass scores without binary cutoff. We also explore clustering, aiming to give insight to product category. Also, different from approaches from previous works, we make the best use of the review summary text, a concise version of the review text, and applied count vectorizer to explore its effectiveness and maximize machine learning performance from it.

## 3. The Amazon Product Review Dataset

The training dataset contains 29189 entries of product reviews from Amazon. Information includes the following: Overall: product rating on a scale of [1-5]; verified: a boolean denoting if the review has been verified by Amazon; reviewTime: time of review; reviewerID: The unique ID of the Amazon reviewer (some have left multiple reviews); asin: Product ID (One product will have many different reviews); reviewerName: Encoding of the Amazon reviewer's username; reviewText: The Amazon review; summary: a concise summary of the reviewText; unixReviewTime: unix time of review; vote: How many people voted this review as being helpful; image: If there is an image, link to the image; style: a dictionary of style informa-

overall	1
verified	TRUE
reviewTime	04 7, 2015
reviewerID	868FE8626DAEF71BAD1E6A92F6D930CB
asin	DCBCF41F4BA5D96EEDEE747EC340A56B
reviewerName	80206880F42975D409B5A5EEDA1D6B1F
reviewText	absolute garbage. measured at 16 feet while not being under tension. i needed 19 feet, so i stretched it. ripped in a bunch of places within 3 days. if it is being sold as a 20 foot hose, it needs to go to 20. buy the rhinoflex. it goes to 20 and stays there.
summary	Don't waste your money, buy Rhino-Flex
unixReviewTime	1428364800
vote	36
image	['https://images-na.ssl-images-amazon.com/images/I/71081-RGFrL..SY88.jpg']
style	{'Package Type:': 'Standard Packaging', 'Style:': '20' Sewer Hose Kit'}
category	automotive

Table 1. Example Entry.

tion (e.g. size of shirt, color of phone) (Only available for some samples); Category: The Amazon product category of the product. Table 1 shows an example entry. The data is uniformly representing all 5 overall ratings as well as all 6 categories as shown in Table 3.

The test dataset contains 4500 new entries with the same fields as the training dataset, but without the overall score.

## 4. Methods

All models are implemented in Scikit-Learn (version 1.2.1), and hyperparameters are in default unless specified. To reduce the time for hyperparameter tuning, 10% of the training dataset was used with grid search cross-validation to find the best hyperparameters for both the vectorizers (to transform text data) and the classifiers. The final model is trained on the entire training dataset with 5-Fold Cross-validation. The Macro F1 score from the prediction of the test dataset was calculated.

Model	Logistic Regression	Macro F1 Score
TfidfVectorizer: max_df=0.75, min_df=3, ngram_range=(1, 2), norm='l2', sublinear_tf=True	max_iter = 500, C=10, penalty='l2'	0.81823
CountVectorizer: max_df=0.5, min_df=1, ngram_range=(1, 2)	max_iter = 500, C=1, penalty='l2'	0.83217
Sentiment Analysis	max_iter = 500, C=0.1, penalty='l2'	0.73758

Table 2. Evaluation of Text Vectorization Methods with a Rating Cutoff=3.

## 4.1. Feature Engineering

### 4.1.1 Overall Rating

For binary classification, the rating is converted to "good" and "bad" based on the cutoffs 1,2,3, and 4. For example, when cutoff=3, all samples with a rating  $\leq 3$  will have label 0, and all samples with a rating  $> 3$  have label 1. The rating is unchanged for multiclass classification.

## 4.2. Choice of Vectorizer

In order for machine learning models to train the text data, the reviewText and summary were transformed into numbers, by using vectorizers. Exploring the data we can find the summary, a lot of times, either contains strong sentiment adjectives like "bad/terrible/great" or directly indicates the rating such as "one/five stars". Therefore, it is likely summary contains more keywords indicating the sentiment and fewer data on the product specification than reviewText.

In light of this observation, three methods were explored for classifications: 1) reviewText with Tfidf Vectorizer; 2) summary with Count Vectorizer; 3) sentiment analysis of summary.

For each vectorizer, a grid search with cross-validation was used to find their best hyperparameters. For sentiment analysis, a pre-trained sentiment analysis model from Transformers (version 4.26.1) from the Hugging Face was used to convert the summary to either "positive" or "negative". Specifically, a pipeline of "sentiment-analysis" with the model "distilbert-base-uncased-finetuned-sst-2-english" was used. For each method, a cutoff of 3 and logistic regression were used to train the model. Table 2 shows that the Macro F1 score using summary with Count Vectorizer performs the best. Therefore, it is chosen as the main feature

Rating	automotive	CDs	grocery	cell phones	sports	toys	Total
1	994	996	996	984	996	991	5997
2	993	996	995	989	993	993	5959
3	970	984	977	971	977	983	5862
4	945	979	964	950	965	966	5769
5	926	957	954	905	949	951	5642

Table 3. Uniform Representation of Data Across Ratings and Categories.

processing method for classifiers.

For clustering, since the aim is to cluster by product categories, the model would need more information on product descriptions. Therefore, a count vectorizer was used to transform the reviewText. The specific hyperparameters are as follows: max\_features=1, max\_df=0.5, min\_df=1, ngram\_range=(1, 2).

### 4.3. Choice of Models

For both binary and multiclass classifications, Logistic Regression, SVM, and Random Forest were used. For classification, KMeans, DBSCAN, and Hierarchical Clustering were used.

## 5. Results and Analysis

Table 6, 7, 8, and 9 shows the best hyperparameters of the three models and the performance scores after running 5-fold cross-validation using the training dataset. Table 4 shows the clustering result.

From the binary classification results, we can see that, in terms of Macro F1 scores, logistic regression performs best for all cutoffs except for cutoff=1 being slightly lower than random forest. However, logistic regression takes considerably less time to train, making it the best candidate to scale when the dataset gets large. Across all models, cutoff=3 has the highest macro F1 score, cutoff=2 and 4 have similar scores, while, cutoff=1 has the lowest. Cutoff=3 is a common cutoff for positive and negative sentiments. It suggests that the models can distinguish the overall sentiment direction better than the extent/intensity of them.

For multiclass classification, all three models perform almost equally with SVM being the slight highest. It is interesting to note from the confusion matrix that all three models consistently predicts a false rating of 1 when the true rating is 2. It may suggest that customers may act harsher on ratings than their actual elaborations.

## 6. Conclusions

Table 5 shows the best scores using the test dataset when submitted to the Kaggle Competition, using the models and hyperparameters from the result section. Overall, it

Model	Hyperparameters	Silhouette score
K Means	n_clusters=6, random_state=42, n_init=10	0.8425511223533916
DBSCAN	eps=0.5, min_samples=5	0.998750656474856
Agglomerative Clustering	n_clusters=6	0.7862139817673409

Table 4. Clustering with Different Algorithms.

Task	Metric	Score	Model
Binary Classification Cutoff 1	Macro F1	0.74062	Logistic Regression
Binary Classification Cutoff 2	Macro F1	0.78174	Logistic Regression
Binary Classification Cutoff 3	Macro F1	0.83217	Logistic Regression
Binary Classification Cutoff 4	Macro F1	0.76781	Logistic Regression
Multiclass Classification	Macro F1	0.53207	Logistic Regression
Clustering	Silhouette	0.99875	DBSCAN

Table 5. Final Test Result (from Kaggle Competition).

is demonstrated that these machine learning models can be utilized to predict amazon product ratings based on reviews.

## References

- [1] Gope, Joy Chandra, et al. "Sentiment Analysis of Amazon Product Reviews Using Machine Learning and Deep Learning Models." 2022 International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE), 2022, <https://doi.org/10.1109/icaeee54957.2022.9836420>.
- [2] T. U. Haque, N. N. Saber, and F. M. Shah, "Sentiment analysis on large scale amazon product reviews," in 2018 IEEE international conference on innovative research and development (ICIRD). IEEE, 2018, pp. 1–6.

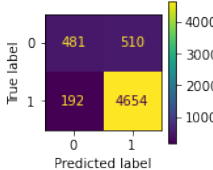
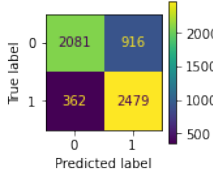
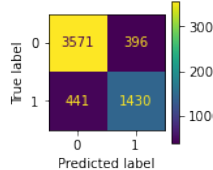
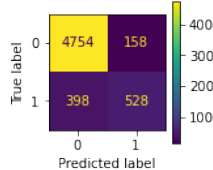
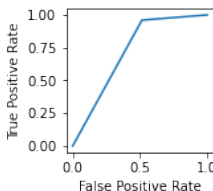
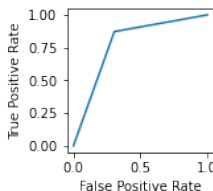
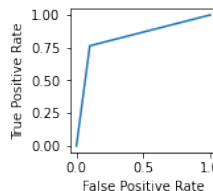
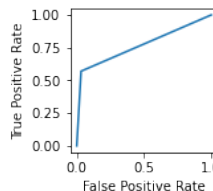
	cutoff=1	cutoff=2	cutoff=3	cutoff=4
Vectorizer	'max_df': 0.5, 'min_df': 1, 'ngram_range': (1, 2)	'max_df': 0.5, 'min_df': 1, 'ngram_range': (1, 2)	'max_df': 0.5, 'min_df': 1, 'ngram_range': (1, 2)	'max_df': 0.5, 'min_df': 2, 'ngram_range': (1, 2)
Logistic Regression	'C': 1, 'penalty': 'l2'	'C': 1, 'penalty': 'l2'	'C': 1, 'penalty': 'l2'	'C': 1, 'penalty': 'l2'
Accuracy	0.879732739420935	0.781089414182939	0.856628982528263	0.904761904761905
Macro F1	0.753997564935065	0.78006723295476	0.834349926468801	0.799920212870877
Confusion Matrix				
AUC Score	0.72287400471349	0.783470552565941	0.83223681152512	0.769014130335371
ROC Curve				

Table 6. Binary Classifier 1: Logistic Regression for Different Cutoffs.

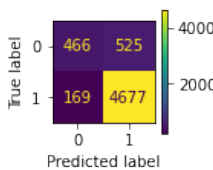
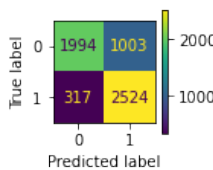
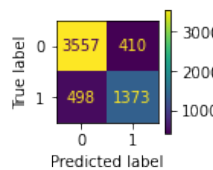
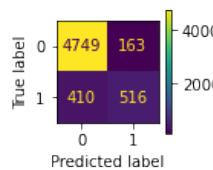
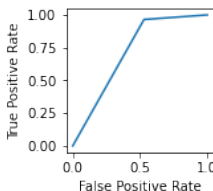
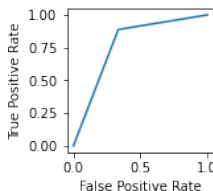
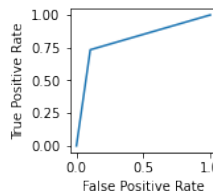
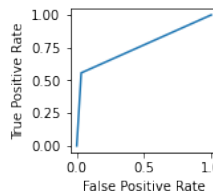
	cutoff=1	cutoff=2	cutoff=3	cutoff=4
Vectorizer	'max_df': 0.5, 'min_df': 1, 'ngram_range': (1, 2)	'max_df': 0.5, 'min_df': 1, 'ngram_range': (1, 1)	'max_df': 0.5, 'min_df': 1, 'ngram_range': (1, 1)	'max_df': 0.5, 'min_df': 1, 'ngram_range': (1, 1)
Logistic Regression	'C': 5, 'gamma': 'scale', 'kernel': 'rbf'	'C': 5, 'gamma': 'scale', 'kernel': 'rbf'	'C': 10, 'gamma': 'scale', 'kernel': 'rbf'	'C': 5, 'gamma': 'scale', 'kernel': 'rbf'
Accuracy	0.881103306493062	0.773895169578623	0.844467283316204	0.901849948612539
Macro F1	0.75205863025987	0.772016165984406	0.819158234395642	0.793047308038163
Confusion Matrix				
AUC Score	0.717678982905581	0.776875784619537	0.81523975793347	0.762025691039179
ROC Curve				

Table 7. Binary Classifier 2: SVM for Different Cutoffs.

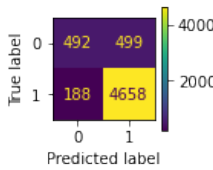
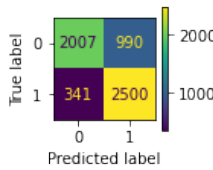
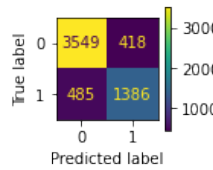
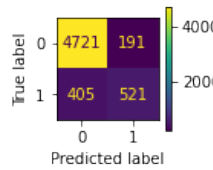
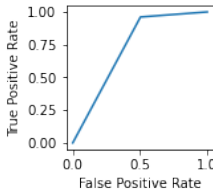
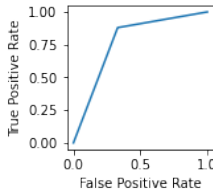
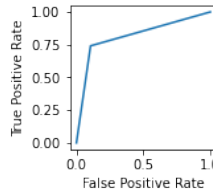
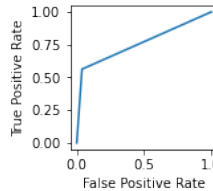
	cutoff=1	cutoff=2	cutoff=3	cutoff=4
Vectorizer	max_df': 1.0, 'min_df': 3, 'ngram_range': (1, 1)	max_df': 0.75, 'min_df': 2, 'ngram_range': (1, 1)	'max_df': 0.75, 'min_df': 3, 'ngram_range': (1, 1)	'max_df': 0.5, 'min_df': 2, 'ngram_range': (1, 2)
Logistic Regression	'max_depth': 2000, min_samples_split: 7, 'n_estimators': 75	'max_depth': 3000, 'min_samples_split': 7, 'n_estimators': 100	'max_depth': 3000, 'min_samples_split': 7, 'n_estimators': 50	'max_depth': 1000, 'min_samples_split': 7, 'n_estimators': 50
Accuracy	0.882302552681172	0.775265501884207	0.845323741007194	0.897910243233984
Macro F1	0.760094772286447	0.773644750927709	0.820712410948632	0.788383629387814
Confusion Matrix				
AUC Score	0.728836665774055	0.778082200468684	0.81770551733792	0.761875312190009
ROC Curve				

Table 8. Binary Classifier 3: Random Forest for Different Cutoffs.

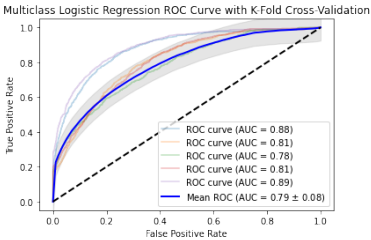
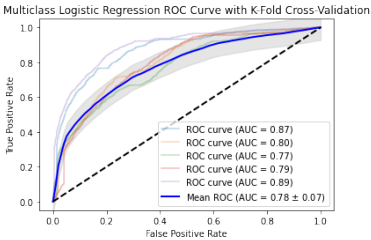
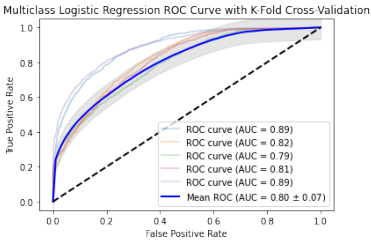
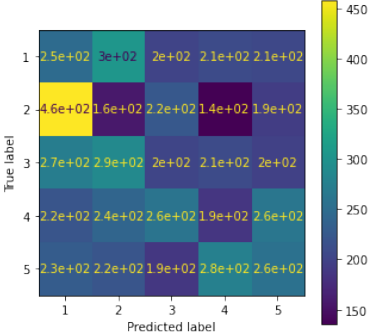
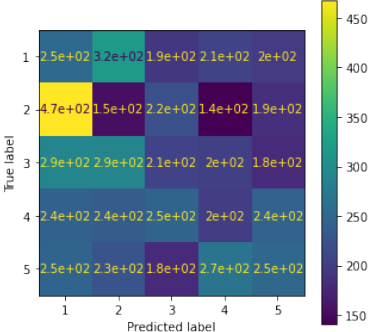
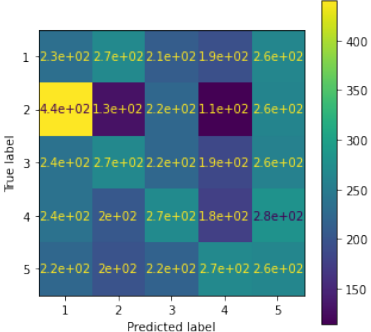
	Logistic Regression	SVM	Random Forest
Model	max_iter = 1000, C=5, penalty='l2', multi_class='ovr', solver='liblinear'	decision_function_shape='ovr', C=1, kernel='linear', probability=True)	Default parameters
Vectorizer	CountVectorizer max_df=0.5, min_df=1, ngram_range=(1, 2)	CountVectorizer max_df=0.5, min_df=1, ngram_range=(1, 2)	CountVectorizer max_df=0.5, min_df=1, ngram_range=(1, 2)
Macro F1	0.47132387431083300	0.472290240419424	0.46644456950292
Accuracy	0.472166337025096	0.473639274047856	0.471515546563214
Auc	0.787829346719452	0.779471159686766	0.797974784488607
ROC Curve			
Confusion Matrix			

Table 9. Multiclass Classifications.