Unlocking the Potential of Amazon Reviews: A Study of Binary and Multiclass Classifiers and Clustering Technique

Sookyoung Park
Dartmouth College, Computer Science Digital Arts(MSDA)
Hanover, New Hampshire
sookyoung.park.gr@dartmouth.edu

Abstract

This study focuses on predicting the overall value of Amazon reviews through binary/multiclass classification models and clustering based on product categories. Three different machine learning models were implemented, including three binary classifications for each of the four cutoffs, three multiclass classifications, and clustering. For binary and multiclass classifications, the models were trained using the features 'reviewText', 'summary', and 'verified'. The models were evaluated using various metrics, such as confusion matrix, ROC, AUC, F1-score, and accuracy. Overall, this study provides insights into the effectiveness of machine learning models for classifying and clustering. The models used in this project can be extended to various other fields, such as sentiment analysis and product recommendation systems, providing valuable insights for businesses and academic institutions.

Keywords: *Machine Learning, Clustering, Classification, Classifiers, Logistic Regression, Decision Tree, Random Forest, Naive Bayes, GaussianNB*

1. Introduction

Online reviews have become an important and meaningful data for consumers in making purchasing decisions. Analyzing these reviews can provide valuable insights into consumer preferences, opinions, and reliability, which can be useful for businesses and academic institutions. With the increasing volume of reviews, it is challenging to extract meaningful information manually. Machine learning has been widely used to automate the process of analysis and prediction of ratings. In this study, we focus on predicting the overall value of Amazon reviewText using binary/multiclass classification models and clustering based on product categories. We implemented three different machine learning models, including three binary classifications for each of the four cutoffs, three multiclass classifications, and clustering. The models were trained using the features 'reviewText', 'summary', and 'verified' and evaluated using various metrics, such as confusion matrix, ROC, AUC, F1-score, and accuracy. The results of our study showed that the binary classification model performed the best with an accuracy of about 90%. The clustering model was able to group the 'reviewText' based on the 'category' with a Silhouette score of approximately 0.7 and a Rand index of 0.2. The findings of this study demonstrate the effectiveness of machine learning models for classifying and clustering reviews. Moreover, the models developed in this project can be extended to other fields, such as sentiment analysis and product recommendation systems, providing valuable insights for businesses and academic institutions.

2. Related Work

This section reviews the literature on sentiment analysis, highlighting the various approaches and models proposed by researchers in this field.

The author in [4] proposes a product recommendation system using machine learning techniques. The method involves constructing recommendation and non-recommendation product databases using consumer information and product information big data. Two different neural network models are implemented based on the database: recommendation product database, non-recommendation product database. Finally, the system provides a final recommendation product by removing the duplicate products from the two sets of recommendation products. The proposed system is designed to provide fast and accurate recommendations to consumers.

The author in [5] aims to develop a model that predicts the success of crowdfunding projects with deep learning. The deep-learning model could provide insights in the pre-launching stage and in the early stage of fundraising using the datasets from Kaggle and historical records of Kickstarter campaigns. The conclusion of the study is that the MLP model has the most favorable outcome with the highest degree of confidence.

The author in [6] suggests a method for clustering and identifying similarities among users of a digital tourism platform based on the sentiments they express in their reviews or comments. The sentiment analysis includes language detection and syntax treatment. To sum up, this study provides a method for exploring the needs and desires of clients based on their digital footprint and can assist in the development and improvement of tourism services and products.

3. Methods

The methodology for each classifier model, including Binary Classification, Multiclass Classification, and Clustering, is organized into three steps. The same train and test dataset will be used for all models. The train dataset contains 13 features, including overall, verified, reviewTime, reviewID, asin, reviewName, reviewText, summary, unixReviewTime, vote, image, style, and category.

3.1. Binary Classification

The data for this study was obtained from Amazon. Three key features were selected to predict the overall score: Verified, reviewText, and summary. The train dataset includes a total of 29,189 data points with 13 features, while the test dataset includes 4500 data points. In the tes data, overall score is not provided.

3.1.1 Data Preprocessing

To effectively train machine learning models using text data, it is important to convert the text data into a numerical format, specifically a sparse matrix format, as most machine learning algorithms require numerical input. This can be achieved using TfidfVectorizer or CountVectorizer, which are essential tools for transforming text data into sparse matrices. Therefore, the conversion of text data into sparse matrix format using these vectorizers is a crucial step in the machine learning process when using summary, reviewText, and verified features for training.

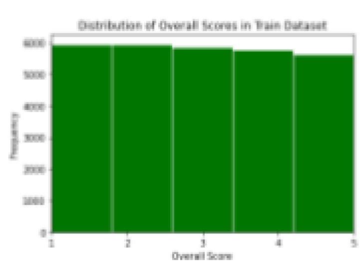

Figure 1: Distribution of overall score in train dataset

3.1.1.1 Verified

As the verified feature has boolean. We need to convert the boolean value(0 or 1) to a sparse matrix.

3.1.1.2 Summary and reviewText

When converting the 'reviewText' feature into a sparse matrix for machine learning, TfidfVectorizer is selected over other methods because it takes into account the importance of each word not only within the specific text sample but also across the entire corpus. TfidfVectorizer is able to capture the semantic meaning of the text data more effectively. Therefore, it is a suitable method for transforming the 'reviewText' feature into a sparse matrix.

3.1.1.3 Merge converted sparse matrix

After converting the 'summary', 'reviewText', and 'verified' features into sparse matrices, they need to be combined into a single matrix to be used as input for the model. To achieve this, the 'hstack' function from the scipy.sparse library is utilized.

3.1.2 Implement Classifiers

Before we apply classifier models, we need to create a function to set a cutoff to make a label. For example, if cutoff=3, all samples with a rating<=3 will have label 0, and all samples with a rating>3 have label 1.

3.1.2.1 Designate a label

To transform the continuous rating scores into discrete labels for the binary classification, the entire dataset is labeled using the previously defined cutoff function. The resulting labels are then used as the target variable for training the classification models.

3.1.2.2 Optimize the quality of Input Dataset

The SelectKBest function is used to select the most important features for the classification models.

3.1.2.3 Initialize a Classifier

3.1.2.3.1 Logistic Regression

- Cutoff=1

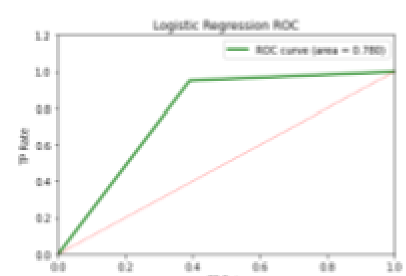

Figure 2: ROC curve of cutoff 1 in Logistic Regression

The highest accuracy was achieved using the tuned hyperparameters: k=1250 for the SelectKBest, C:45, solver: saga, and max_iter: 70.

Confusion Matrix :
[[1110 750]
[337 6595]]
AUC : 0.779802029
F1 Score: 0.802309038
Accuracy : 0.879867534

- Cutoff=2

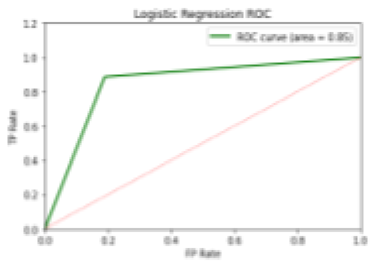

Figure 3: ROC curve of cutoff 2 in Logistic Regression

The highest accuracy was achieved using the tuned hyperparameters: k=8000 for the SelectKBest, C:45, solver: saga, and max_iter: 63.
Confusion Matrix :
[[2917 681]
[[587 4572]]
AUC : 0.848473220838
F1 Score: 0.8498280927
Accuracy : 0.855201553

- Cutoff=3

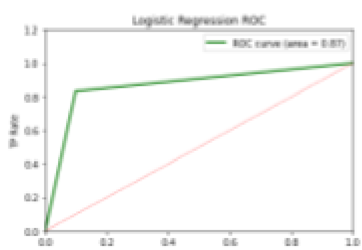

Figure 4: ROC curve of cutoff 3 in Logistic Regression

The highest accuracy was achieved using the tuned hyperparameters: k=12000 for the SelectKBest, C:64, solver: saga, and max_iter: 90.
Confusion Matrix:
[[4859 530]
[563 2805]]
AUC : 0.86724489
F1 Score: 0.867918976
Accuracy : 0.875185565

- Cutoff=4

The highest accuracy was achieved

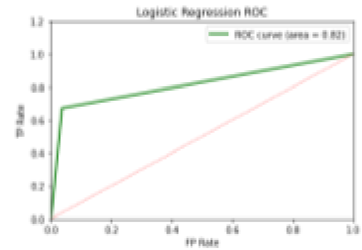

Figure 5: ROC curve of cutoff 4 in Logistic Regression

using the tuned hyperparameters: k=2500 for the SelectKBest, C:19, solver: saga, and max_iter: 77.
Confusion Matrix:
[[6838 256]
[546 1117]]
AUC : 0.8177954
F1 Score: 0.84022111
Accuracy : 0.908416124

3.1.2.3.2 Decision Tree

- cutoff=1

The highest accuracy was achieved using the tuned hyperparameters:

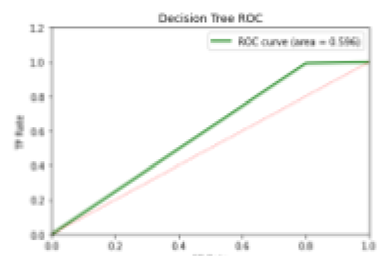

Figure 6: ROC curve of cutoff 1 in Decision Tree

k=20802 for the SelectKBest, criterion:gini, max_depth:23, min_samples_split: 5
Confusion Matrix:
[[362 1463]
[40 6892]]
AUC : 0.5962929119
F1 Score: 0.613391099
Accuracy : 0.828365878

- cutoff=2

The highest accuracy was achieved using the tuned hyperparameters:

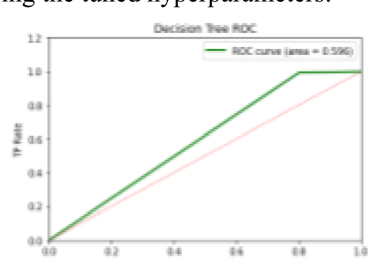

Figure 7: ROC curve of cutoff 2 in Decision Tree

k=8000 for the SelectKBest,
criterion:gini, max_depth:32,
min_samples_split: 5

Confusion Matrix:
[[2256 1342]
[1014 4145]]
AUC : 0.7152326
F1 Score: 0.71782801
Accuracy : 0.730958096

- cutoff=3
The highest accuracy was achieved
using the tuned hyperparameters:

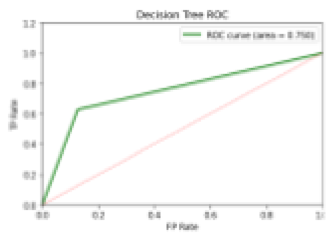

Figure 8: ROC curve of cutoff 3 in Decision Tree

k=8000 for the SelectKBest,
criterion:gini, max_depth:23,
min_samples_split: 2
Confusion Matrix:
[[4702 687]
[1255 2113]]
AUC: 0.74994669
F1 Score: 0.75699375
Accuracy: 0.778234555

- cutoff=4
The highest accuracy was achieved
using the tuned hyperparameters:

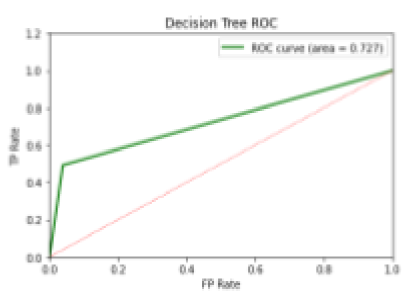

Figure 9: ROC curve of cutoff 4 in Decision Tree

k=8000 for the SelectKBest,
criterion:gini, max_depth:10,
min_samples_split: 3
Confusion Matrix:
[[6816 278]
[844 819]]
AUC: 0.7266477
F1 Score: 0.7587155
Accuracy: 0.8718739

3.1.2.3.3 Random Forest

- cutoff=1
The highest accuracy was achieved
using the tuned hyperparameters:
k=8000 for the SelectKBest,

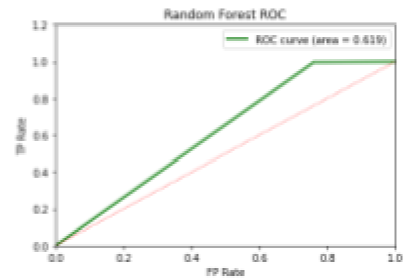

Figure 10: ROC curve of cutoff 1 in Random Forest

n_estimators:1300, max_depth:,
min_samples_split: 20,
min_samples_leaf: 2
Confusion Matrix:
[[439 1386]
[23 6909]]
AUC: 0.618614999
F1 Score: 0.64568802
Accuracy: 0.8391001

- cutoff=2
The highest accuracy was achieved
using the tuned hyperparameters:
k=8000 for the SelectKBest,

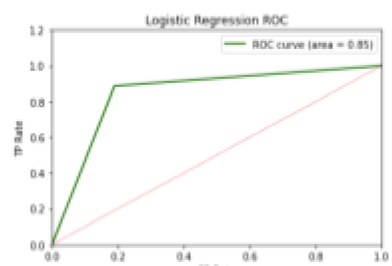

Figure 11: ROC curve of cutoff 2 in Random Forest

n_estimators:500, max_depth:150,
min_samples_split: 10,
min_samples_leaf: 1
Confusion Matrix:
[[2481 1117]
[436 4723]]
AUC: 0.8025186
F1 Score: 0.810216
Accuracy: 0.82265

- cutoff=3
The highest accuracy was achieved
using the tuned hyperparameters: k=800
for the SelectKBest, n_estimators: 100,
max_depth:15, min_samples_split: 2,

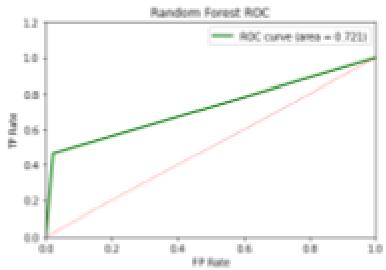

Figure 12: ROC curve of cutoff 3 in Random Forest

min_samples_leaf: 1

Confusion Matrix:
[[5275 114]
 [1805 1563]]
AUC: 0.7214597
F1 Score: 0.732860
Accuracy: 0.780861

- cutoff=4
  The highest accuracy was achieved using the tuned hyperparameters: k=100

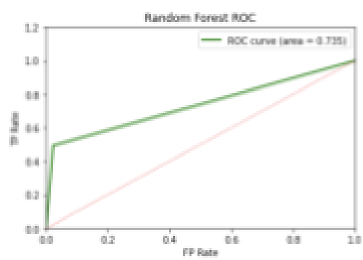

Figure 13: ROC curve of cutoff 4 in Random Forest

for the SelectKBest, n_estimators:300, max_depth: 80, min_samples_split: 7, min_samples_leaf: 4
Confusion Matrix:
[[6933 161]
 [844 819]]
AUC:0.734894
F1 Score: 0.776084
Accuracy: 0.88523

3.2. Multiclass Classification

The data for this study was obtained from Amazon. Three key features were selected to predict the overall score: Verified, reviewText, and summary. The train dataset includes a total of 29,189 data points with 13 features, while the test dataset includes 4500 data points. In the tes data, overall score is not provided.

3.2.1 Data Preprocessing

To effectively train machine learning models using text data, it is important to convert the text data into a numerical format, specifically a sparse matrix format, as most machine learning algorithms require numerical input. This can be achieved using TfidfVectorizer or CountVectorizer, which are essential tools for transforming text data into sparse matrices. Therefore, the conversion of text data into sparse matrix format using these vectorizers is a crucial step in the machine learning process when using summary, reviewText, and verified features for training.

3.2.1.1 Verified

As the verified feature has boolean. We need to convert the boolean value(0 or 1) to a sparse matrix.

3.2.1.2 Summary and ReviewText

When converting the 'reviewText' feature into a sparse matrix for machine learning, TfidfVectorizer is selected over other methods because it takes into account the importance of each word not only within the specific text sample but also across the entire corpus. TfidfVectorizer is able to capture the semantic meaning of the text data more effectively. Therefore, it is a suitable method for transforming the 'reviewText' feature into a sparse matrix.

3.2.1.3 Merge converted sparse matrix

After converting the 'summary', 'reviewText', and 'verified' features into sparse matrices, they need to be combined into a single matrix to be used as input for the model. To achieve this, the 'hstack' function from the scipy.sparse library is utilized.

3.2.2 Implement Classifiers

Before we apply classifier models, we need to create a function to set a target class: 1,2,3,4,5.

3.2.2.1 Logistic Regression

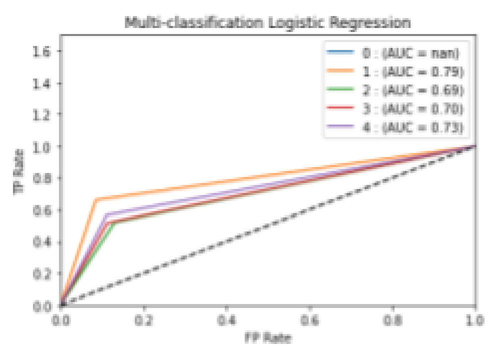

Figure 14: ROC curve of multiclass classification in Logistic Regression

The highest accuracy was achieved using the tuned hyperparameters: k=11000 for the SelectKBest, C:45, solver: saga, max_iter:70.
Confusion Matrix:
[[1201 393 134   55   42]

[363 919 307 133   51]
[132 340 922 304   93]
[56 130 243 966 310]
[38   47 101 292 1185]]
F1 Score: 0.59396115
Accuracy: 0.59301130

3.2.2.2 Decision Tree

The highest accuracy was achieved using the

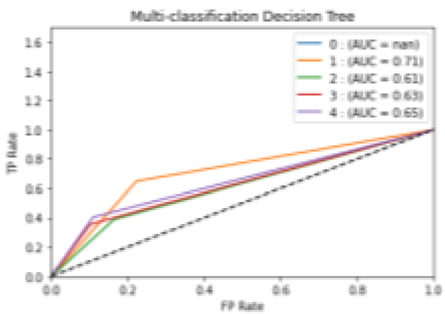

Figure 15: ROC curve of multiclass classification in Decision Tree

tuned hyperparameters: k=5000 for the SelectKBest, criterion: gini, max_depth: 25, min_samples_split: 2, min_samples_leaf: 1

Confusion Matrix:

[[1182 327 146   98   72]
[590 689 248 166   80]
[370 412 637 271 101]
[272 268 220 688 257]
[314 140   99 233 877]]
F1 Score: 0.463406
Accuracy: 0.465113

3.2.2.3 Random Forest

The highest accuracy was achieved using the tuned hyperparameters: k=8500 for the

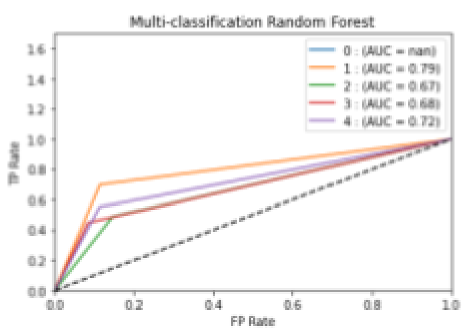

Figure 16: ROC curve of multiclass classification in Random Forest

SelectKBest, n_estimators:1300, max_depth: None, min_samples_split: 4, min_samples_leaf: 3

Confusion Matrix:

[[1285 362   58   59   61]
[484 826 250 138   75]
[152 417 816 310   96]
[82 173 236 924 290]
[83   89   58 287 1146]]
F1 Score: 0.5694991
Accuracy: 0.5706292

3.3. Clustering

The data for this study was obtained from Amazon, and only two features were selected for use in the test dataset, namely 'reviewText' and 'category.' The study will use these two features to analyze and draw insights from the data.

3.3.1 Data Analysis Process

In this study, CountVectorizer was utilized with the parameters stop_words and lowercase. These parameters were chosen to enhance the accuracy of the clustering by removing stop words and converting all text to lowercase. And convert it to a sparse matrix for clustering.

3.3.2 Implement Clustering

3.3.2.1 K-means Clustering

In order to achieve the highest accuracy score, a KMeans clustering model was utilized with 5 distinct classes and a random_state value of 40.

Silhouette score: 0.7233433128592512
Rand index: 0.17034521251636167

4. Results and Analysis

4.1. Model Comparison

4.1.1 Binary Classification

4.1.1.1 Terms of assessment criteria

For binary classification models, various evaluation metrics were used including confusion matrix, AUC, F1 Score, and Accuracy to assess the performance of the models.

4.1.1.2 Assessment

Based on the chart, the Logistic Regression model achieved the highest score across all evaluation criteria. On the other hand, the Decision Tree model had the lowest scores across all evaluation criteria.

4.1.1.3 Cutoff=1

	AUC	F1 Score	Accuracy
Logistic Regression	0.779802029	0.802309038	0.879867534
Decision Tree	0.596292911	0.613391099	0.828365878
Random Forest	0.618614999	0.64568802	0.8391001

Figure 17: Assessment Comparison on Binary Classification in cutoff 1

4.1.1.4 Cutoff=2

	AUC	F1 Score	Accuracy
Logistic Regression	0.8484732	0.84982809	0.855201553
Decision Tree	0.7152326	0.71782801	0.730958096
Random Forest	0.8025186	0.810216	0.82265

Figure 18: Assessment Comparison on Binary Classification in cutoff 2

4.1.1.5 Cutoff=3

	AUC	F1 Score	Accuracy
Logistic Regression	0.86724489	0.867918976	0.876185565
Decision Tree	0.74994669	0.75699375	0.7778234555
Random Forest	0.7214597	0.732860	0.780861

Figure 19: Assessment Comparison on Binary Classification in cutoff 3

4.1.1.6 Cutoff=4

	AUC	F1 Score	Accuracy
Logistic Regression	0.8177954	0.84022111	0.908416124
Decision Tree	0.7266477	0.7587155	0.8718739
Random Forest	0.734894	0.776084	0.88523

Figure 20: Assessment Comparison on Binary Classification in cutoff 4

4.1.2 Multiclass Classification

4.1.2.1 Terms of assessment criteria

For binary classification models, various evaluation metrics were used including confusion matrix, AUC, F1 Score, and Accuracy to assess the performance of the models.

4.1.2.2 Assessment

Based on the chart, the Logistic Regression model achieved the highest score across all evaluation criteria. On the other hand, the Decision Tree model had the lowest scores across all evaluation criteria.

	Score	Accuracy
Logistic Regression	0.59396115	0.59301130
Decision Tree	0.463406	0.465113
Random Forest	0.5694991	0.5706292

Figure 21: Assessment Comparison on Multiclass Classification

5. Conclusion

This thesis aimed to predict the overall value of Amazon reviews through binary/multiclass classification models and clustering based on product categories. The study is focused on three machine learning models and evaluated them using various metrics. The results provide insights into the effectiveness of these models for classification and clustering, with potential applications in sentiment analysis and product recommendation systems. The study also suggests the models can be applied in other domains, offering valuable insights for academic institutions and businesses.

References

[1] ChatGPT

[2] Ansh Gupta, Aryan Rastogi, and Avita Katal, Feb. 2022, “A Comparative Study of Amazon Product Review Using Sentiment Analysis

[3] Yi Luo and Xiaowei Xu, Aug.2019, “Predicting the Helpfulness of Online Restaurant Reviews Using Different Machine Learning Algorithm: A case study of Yelp”

[4] Hong Seung U, 2022, “Product Recommendation System Using Machine Learning Technique And Method Thereof”

[5] Pi-Fen Yu, Fu-Ming Huang, Chuan Yang, Yu-Hsin Liu, Zi-Yi Li, and Cheng-Hung Tsai, 2018, “Prediction of Crowdfunding Project Success with Deep Learning”

[6] Sandra Jardim and Carlos Mora, 2021, “Customer Reviews Sentiment-Based Analysis and Clustering for Market-Oriented Tourism Services and Products Development or Positioning”