Amazon Product Review Classification and Clustering Using Machine Learning

Digvijay Yadav
MS Quantitative Biomedical Science. Dartmouth College
Hanover, United States
digvijay.yadav.gr@dartmouth.edu

***Abstract*—I implemented binary classification, multi-class classification, and clustering-based machine learning approaches to analyze and categorize product reviews in Python, for the amazon product review dataset. The models achieved an overall macro F1 score of 0.719 for Baseline 1, 0.782 for Baseline 2, 0.807 for Baseline 3, and 0.714 for Baseline 4 of Binary Classification. Additionally, the model achieved an overall macro F1 score of 0.473 in Multiclass Classification and using K-Means, 0.64 Silhouette score for clustering.**

I. Related Work

Previous works have revolved around sentiment analysis of the reviews using context-based approaches as is in the paper [1]. Others involved in using pretrained deep learning models like CNN, and BERT on larger datasets.

II. Methods

Our analysis involved experimenting with different machine-learning techniques to analyze the text-based data of the amazon review dataset. I started with the cardinal step of machine learning, i.e., Data preprocessing and exploratory data analysis. I then worked on implementing machine learning algorithms to analyze the results. For the preprocessing and modeling pipeline, I used the Sklearn library only. For Binary Classification and Multiclass classification task, we had to beat the baseline scores on the Kaggle Competitions. The target scores for threshold 1 and 4 was 0.70, for 2, it was 0.78, and for 3 it was 0.80. For the multiclass classification, it was 0.47. The scoring parameter was F1-macro.

A. Data Cleaning and Preprocessing

As is with all the real-world data, our dataset was of around 29000 samples consisting of an overall column in the training data, and other text-based columns like review text, Summary, Category, etc. Since the project was divided into three sections involving Binary Classification, Multi-Class classification, and clustering approaches, the preprocessing pipeline had to be tuned a little to meet the specifications of the task at hand. The overall approach involved identifying missing values, checking for data imbalance, and applying text preprocessing algorithms like removing whitespace, punctuations, stop-words, and, alpha-numeric digits.

For Binary Classification, I further processed data by creating a new column called product review, based on the overall column of the training data. I started with thresholds 1,2,3 and 4. There was a moderate data imbalance observed in the product review based on the binary class labels created in the previous steps. The imbalance in the dataset was resolved by including the parameter of class weight as balanced in the algorithms used for modeling. For Multiclass and Clustering, the product review column was not considered in the analysis. For the analysis, I considered using the review text column as it had information regarding the product, whether it was a good product or bad. After preprocessing the review text column by removing stop words I used Count Vectorizer, and TFIDF Vectorizer to transform the text data into numerical representations that the machine learning model would be able to understand easily. For binary classification thresholds of 1 and 4, and for the Clustering task, I used the Count Vectorizer transformation, for 2, 3, and Multiclass classification I used Tf-IDF vectorization.

B. Logistic Regression

Logistic Regression or Logit regression is a statistical framework used for understanding the relationship like Binary classification and one vs rest-based Multiclass classification. The model is extremely generalizable and used the sigmoid function to compute the probabilities. I used it for all 4 Binary Classification tasks and Multiclass classification tasks. After experimenting with various hyperparameters, I used the regularization value of 1, L2 normalization, and class weights as balanced to achieve baseline scores for Baselines 1, 3, and 4 of Binary Classification. The regularization value of 0.6, along with the same other parameters worked better for beating Baseline 2. For the Multiclass classification task, I used a regularization parameter of 3, along with a balanced class weight, and L2 penalty.

C. Decision Tree Classifier

I experimented with other algorithms like Decision Trees, because of their generalizability and faster computations even for larger datasets, I used this model because of my prior experience with training this model on Titanic, a Breast Cancer classification dataset, both in which the model had performed better. I experimented with criterion, and splitter hyperparameter, as they had better results and were computationally faster as well.

D. Random Forest Classifier

Random Forest is an extremely powerful ensembling technique that requires very limited hyperparameter tuning to deliver wonderful results, it is a commonly used algorithm in Data Science related use cases. It is basically computed by bagging many decision trees together and training each of them with different training samples. The evaluation is done based on the cumulative votes of the decisions made by each decision tree. It is an effective algorithm that reduces overfitting and is extremely generalizable. I used this model for Baselines 1, and 2 of the Binary Classification task.

E. Extra Tree Classifier

Extra Tree Classifier is an extremely randomized tree classifier that is an ensemble learning method, it is a variant of Decision Tree Classifier, that is more computationally efficient and can handle large dimensional datasets. The algorithm works in a way that it randomly splits the nodes, and creates decision trees on a random subset of training data. This approach with the random creation of decision trees helps in addressing overfitting and improves the accuracy of the model. Extra Tree Classifier models were used in Baselines 3 and 4 tasks of Binary Classification.

F. Multinomial Naive Bayes

Used predominantly in text-based machine learning analysis, Multinomial Naive Bayes is a probabilistic classification algorithm. It is a more efficient approach in terms of computation and is effective with all kinds of text data, including big datasets. The goal of Multinomial Naive Bayes is to forecast the likelihood of each class given the features. The procedure initially determines the prior probability of each class, which is the probability of each class in the training dataset. After determining the class for each feature, the computer calculates the conditional probability of each feature. This is accomplished by dividing the overall frequency of the feature across all classes in the training dataset by the frequency of the feature across all classes in the training samples.

G. Ridge Classifier

It involves using L2 norm regularization parameter to prevent overfitting, This classifier usually converts the data types in the range of -1, 1 before it applies regression techniques to it. It is different from Logistic Regression in that it adds a regularization parameter in the loss function that is L2 inspired. It also reduces the parameters that are not helpful in analysis to zero, thereby making the model effective against outliers. It is a computationally efficient algorithm, I used it for Multiclass classification tasks.

H. K-Means

K Means is an unsupervised algorithm that iteratively partitions the data points into a K group of clusters until there is no change in the centroids. This process continues until the centroids don't change. The clustering quality of the model is evaluated using the Silhouette score. The data points are assigned to clusters based on the minimum squared distances between the data points and centroids.

III. Results and Analysis

A. Binary Classification 1

For this task, the threshold for product review was 1, so all the reviews that were more than or equal to 1 were labeled as 1, and the rest were labeled as 0. I used Logistic Regression, Decision Trees, and Random Forest classification models. The overall F1 macro score for the Logistic Regression model was 0.707, the AUC score was 0.74, the Accuracy was 0.80, and the weighted average was 0.80. The overall F1 macro score for the Decision Tree model was 0.631, the Accuracy was 0.76, and the weighted average was 0.76. The overall F1 macro score for the Random Forest model was 0.62, the Accuracy was 0.81, and the weighted average was 0.78.

B. Binary Classification 2

For this task, the threshold for product review was 2, so all the reviews that were more than or equal to 2 were labeled as 1, and the rest were labeled as 0. I used Logistic Regression, Decision Trees, and Random Forest classification models. The overall F1 macro score for the Logistic Regression model was 0.76, the AUC score was 0.78, the Accuracy was 0.79, and the weighted average was 0.79. The overall F1 macro score for the Decision Tree model was 0.656, an AUC score of 0.66, the Accuracy was 0.67, and the weighted average was 0.68. The overall F1 macro score for the Random Forest model was 0.704, an AUC score of 0.69, the Accuracy was 0.71, and the weighted average was 0.71.

C. Binary Classification 3

For this task, the threshold for product review was 3, so all the reviews that were more than or equal to 3 were labeled as 1, and the rest were labeled as 0. I used Logistic Regression, Decision Trees, and Extra Tree classification models. The overall F1 macro score for the Logistic Regression model was 0.792, the AUC score was 0.81, the Accuracy was 0.83, and the weighted average was 0.83. The overall F1 macro score for the Decision Tree model was 0.689, the AUC Score was 0.698, the Accuracy was 0.71, and the weighted average was 0.71. The overall F1 macro score for the Extra Tree Classifier model was 0.644, the AUC score was 0.675, the Accuracy was 0.69, and the weighted average was 0.69.

D. Binary Classification 4

For this task, the threshold for product review was 4, so all the reviews that were more than or equal to 4 were labeled as 1, and the rest were labeled as 0. I used Logistic Regression, Decision Trees, and Extra Tree classification models. The overall F1 macro score for the Logistic Regression model was 0.72, the AUC score was 0.78, the Accuracy was 0.81, and the weighted average was 0.83. The overall F1 macro score for the Decision Tree model was 0.656, the Accuracy was 0.77, and the weighted average was 0.78. The overall F1 macro score for the Extra Tree Classifier model was 0.627, the AUC score was 0.64, the Accuracy was 0.77, and the weighted average was 0.77.

E. Multiclass Classification

For this task, I used logistic regression, Multinomial Naive Bayes Algorithm, and Ridge Classifier algorithms. The overall F1 macro score for the Logistic Regression model was 0.472, the AUC score was 0.68, and the weighted average was 0.48. The overall F1 macro score for the Multinomial Naive Bayes model was 0.45, The AUC score was 0.66, and the weighted average was 0.46. The overall F1 macro score for the Ridge Classifier was 0.467, the AUC was 0.68, and the weighted average was 0.48.

F. Clustering

Using the KMeans algorithm we evaluated the model on the Silhouette score and Rand index, after processing the text data using Count Vectorizer, I obtained a silhouette score of 0.638, and a rand score of 0.407.

IV. Conclusions

The logistic regression model was the best-performing model amongst the models I worked on for this classification task, and with some fine-tuning of hyperparameters, the results were excellent, to the point of beating the baseline scores in the Kaggle competition for Binary and Multiclass classification.

V. References

References

[1] Chinnalagu A, Durairaj AK. Context-based sentiment analysis on customer reviews using machine learning linear models. PeerJ Comput Sci. 2021 Dec 17;7:e813. doi: 10.7717/peerj-cs.813. PMID: 35036535; PMCID: PMC8725657.

[2] https://scikit-learn.org/stable/index.html.

[3] Alharbi et al. (2021) Alharbi NM, Alghamdi NS, Eman HA, Ali Amri JF. Evaluation of sentiment analysis via word embedding and RNN variants for Amazon online reviews. Hindawai, Mathematical Problems in Engineering. 2021;2021:5536560.

[4] https://pandas.pydata.org/.