Text Classification Analysis on Amazon Food Reviews: Compare the Accuracy and AUC of Machine Learning Models to discover the better Model as the NLP Model for Reviews

Huaqing Gu / Mingcan Li / Qishan Zhao / Wenjie Chen

1. Problem definition:

This project is aiming for an in-depth investigation of Amazon food reviews. Customer names, rating scores, and text of reviews of different products are collected to form our dataset "Reviews.csv". We will use this dataset to classify the emotional patterns in customers' reviews, and build a prediction model where we will be able to predict whether a review is positive or negative. In this process, we will test with different models and figure out a better one for the reviews. We need to answer the following questions in our projects:

- How many machine learning algorithms can we use to train the NLP model?
- Which model presents better results for the reviews dataset?
- What characteristics are we utilizing to define the results?
- What are limitations that could occur in the process of choosing a better model?

2. Description of background:

Comments are very important for all the customers in online shopping, especially after the pandemic, more and more people relied on the online window shopping rather than offline shopping. One of the biggest online shopping platforms used by us is Amazon. The reviews on the products highly affected the customers' decisions. For example, great reviews would increase the number of purchases. Therefore, the analysis of comments would be extremely helpful for companies to analyze customer behavior of products in certain areas.

According to our analysis, we want to identify the best way for companies to analyze what kind and which product would customers be more willing to purchase. This model will be beneficial for the company in multiple areas such as advertising, customer service and product management. In online shopping, customers are not able to test things before they buy them, like in Norhaslinda Kamaruddin(2021)'s comparative study in Turkish Journal of Computer and Mathematics Education: "(Customers) rely on the information given by the seller and previous customers' ratings to make their decision." Therefore, our main goal is to choose better machine learning algorithms for the NLP models for Amazon Food Reviews and our mission is to help not only the company but also the customers to find suitable products.

3. Description of dataset:

For this dataset, we have the reviews of Amazon Food sections from Oct 1999 - Oct 2012. It contains 568,454 different reviews of 74,258 products coming from 256,059 users. About 260 users who are repeat customers contributed more that 50 reviews each.

4. Description of methods used:

Part 1: Data Preprocess

- **Categorise** (self-defined function)

  We define a function called "*categorise*" to help set labels for each text content based on its review score. If its score is smaller than 3, it is labeled as "negative"; if its score is bigger than 3, it is labeled as "positive". If its score is equal to 3, we ignore it.

- **Random**

  We import *random* functions to randomly select a specific number of text data rows, ensuring the randomness of data selection and improving the imbalanced dataset meanwhile.

- **Natural Language Toolkit (nltk) - stopwords**

  There are lots of commonly used words (such as "the", "is", "a") in the reviews, which are called stop words in natural language processing. We don't want them to take up space in our database, or waste processing time. *NLTK* has a list of stopwords, which can be used directly.

- **Nltk.stem**

  Stemming is the process of producing morphological variants of a root/base word. Since the comments contain different tenses and forms of one word, we import *PorterStemmer* in *Nltk* package to do stemming and reduce redundancy.

- **WordClouds, show_wordcloud** (self-defined function)

  A word cloud can show top frequent words in a text content. The size of each word suggests the frequency and importance of that word. We import *WordCloud* function and self-defined *show_wordcloud* function to visualize the one-word wordclouds for both positive and negative classes.

- **Sklearn.model_selection - Train_test_split**

  The accuracy of a model should be tested on a new/unseen dataset. So we import *train_test_split* to help split the original big data set into two pieces randomly — a training set and a testing set. And the split ratio is 8:2.

Part 2: Converting text to vector

- **Sklearn.feature_extraction.text - CountVectorizer**

To find frequent words for building the bag of words, we need to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text. *CountVectorizer* will create a matrix in which each unique word is represented by a column of the matrix, and each row represents each text sample. The value of each cell suggests the count of the word in that particular text sample.

- **Sklearn.feature_extraction.text - TfidfVectorizer**

Term Frequency Inverse Document Frequency (TF-IDF) is a common algorithm to transform text into a meaningful representation of numbers. TF is the number of times a term appears in a particular document. IDF is a measure of how common or rare a term is across the entire corpus of documents. Higher TF-IDF value means more relevant the term is in that document. We import *TfidfVectorizer* from *Sklearn* to convert a collection of raw text to a matrix of TF-IDF features.

Part 3: Machine learning model

- **Sklearn.linear_model - LogisticRegression**

Logistic regression can describe and estimate the relationship between one dependent binary variable and independent variables. It predicts the probability of occurrence of a binary event utilizing a logit function. Since our label is binary, we import *LogisticRegression* form *Sklearn* to train a logistic regression model on different size of bag of words and TF-IDF, and then use the model to predict the sentiment of a text sample.

- **Sklearn.linear_model - RandomForestClassifier**

The random forest is a collection of decision trees that are associated with a set of bootstrap samples that are generated from the original data set, and then It collects the votes from different decision trees to decide the final prediction. We also implement RandomForestClassifier from Sklearn to predict the sentiment classification based on three kinds of bag of words and TF-IDF.

Part 4: Comparison

- Sklearn.metrics-recall_score,accuracy_score,confusion_matrix,f1_score, precision_score

To get specific conditions of prediction results, we implement performance metrics like recall, accuracy, precision, f1-score and confusion_matrix to see the results distribution.

- Sklearn.metrics - roc_curve, auc

ROC is a probability curve and AUC represents the degree or measure of separability. The Higher the AUC, the better the model is at distinguishing two classes.

Part 5: Visualization

- **Matplotlib**

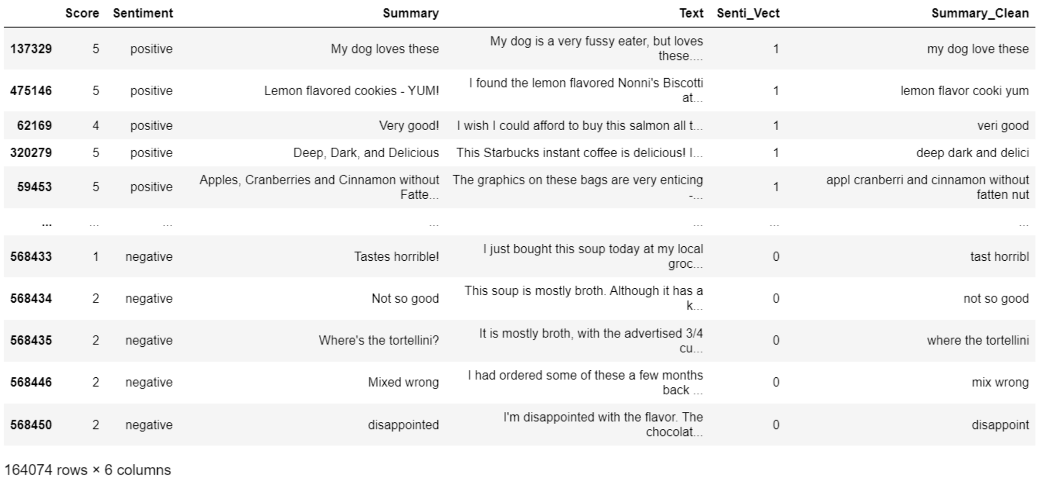
Matplotlib is a visualization library in Python for 2D plots of arrays.

- **ELI5**

ELI5 can visualize and debug various Machine Learning models using a unified API. We implement it to show the model weights of the logistic regression model.

5. Experiment Steps:

At the beginning, we preprocess the data by categorizing the data in 2 different groups and add in sentiment columns, if the score attached to the comments is bigger than 3 is positive, otherwise it is negative. Then, we utilize the nltk tool to clear the stopwords and produce wordclouds to show the frequency words in the dataset. It appears the data itself is highly imbalanced ("positive":443777 rows, "negative": 82037 rows). So we use random selection to reduce the imbalance of the dataset and make two classes have equal size, as shown in Figure 1.

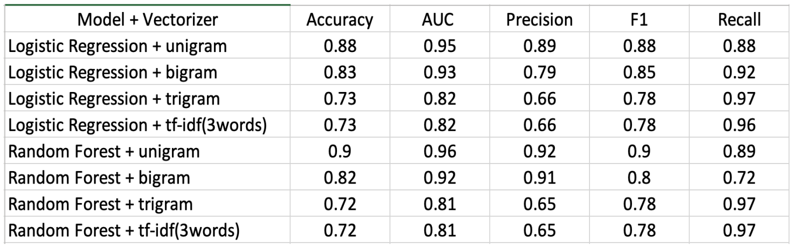


	Score	Sentiment	Summary	Text	Senti_Vect	Summary_Clean
137329	5	positive	My dog loves these	My dog is a very fussy eater, but loves these....	1	my dog love these
475146	5	positive	Lemon flavored cookies - YUM!	I found the lemon flavored Nonni's Biscotti at...	1	lemon flavor cooki yum
62169	4	positive	Very good!	I wish I could afford to buy this salmon all t...	1	veri good
320279	5	positive	Deep, Dark, and Delicious	This Starbucks instant coffee is delicious! I...	1	deep dark and delici
59453	5	positive	Apples, Cranberries and Cinnamon without Fatte...	The graphics on these bags are very enticing -...	1	appl cranberri and cinnamon without fatten nut
...	...	...	...	...	...	...
568433	1	negative	Tastes horrible!	I just bought this soup today at my local groc...	0	tast horribl
568434	2	negative	Not so good	This soup is mostly broth. Although it has a k...	0	not so good
568435	2	negative	Where's the tortellini?	It is mostly broth, with the advertised 3/4 cu...	0	where the tortellini
568446	2	negative	Mixed wrong	I had ordered some of these a few months back ...	0	mix wrong
568450	2	negative	disappointed	I'm disappointed with the flavor. The chocolat...	0	disappoint

164074 rows × 6 columns

Figure1:Equal Final Dataset

Then we use two different machine learning techniques, one is logistic regression and the other is random forest, to process the dataset and calculate the ROC, AUC, accuracy, precision, F1 score and recall scores. For both of the algorithms, we test unigram (one word), bigram(two words), trigram(three words) and tf-idf(three words). Then we compare them together for all the features, like Figure 2:



Model + Vectorizer	Accuracy	AUC	Precision	F1	Recall
Logistic Regression + unigram	0.88	0.95	0.89	0.88	0.88
Logistic Regression + bigram	0.83	0.93	0.79	0.85	0.92
Logistic Regression + trigram	0.73	0.82	0.66	0.78	0.97
Logistic Regression + tf-idf(3words)	0.73	0.82	0.66	0.78	0.96
Random Forest + unigram	0.9	0.96	0.92	0.9	0.89
Random Forest + bigram	0.82	0.92	0.91	0.8	0.72
Random Forest + trigram	0.72	0.81	0.65	0.78	0.97
Random Forest + tf-idf(3words)	0.72	0.81	0.65	0.78	0.97

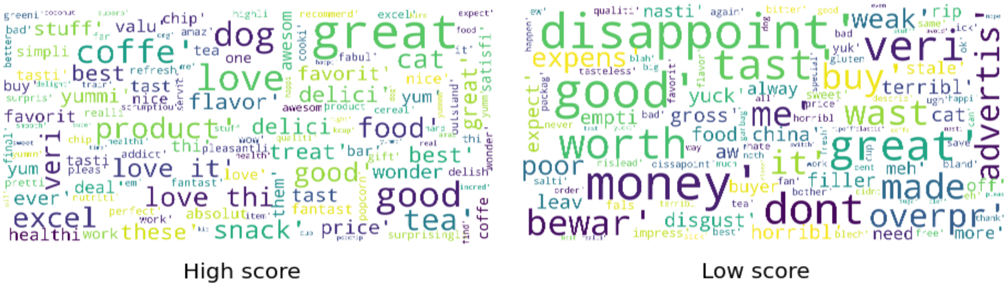
Figure2:Table of Model Comparison

Finally, we analyze the accuracy score of each model and choose a better one for the NLP model for the Amazon Food Reviews.

6. Observations:

Frequent words

First, we get frequent words from original text content before handling the imbalanced dataset from *WordClouds* function for positive and negative text separately. As shown in Figure 3, for positive content, the most frequent words are “delici”, “yummi”, “great product”, “yum”, “love it”, “great” etc.. For negative content, the words of high frequency are “disappoint”, “yuck”, “not good”, “disgust”, “aw”, “horrible” etc..



High score

Low score

Figure3:WordClouds of Amazon Food Reviews

Second, after randomly selecting the same number of positive and negative samples, we used ELI5 to visualize the model weights of four logistic regression models based on unigram, bigram and trigram and three-word tf-idf. We got different results like in Figure 4 (next page).

After carefully observation, we found some interesting conclusions:

(1) Comparing word clouds with model weights:

The weight rankings of bags of words in the logistic regression models are different from the rankings of frequent words in the word clouds, but most of the original frequent words in word clouds still get heavy weight and play an important role in the models.

(2) Comparing four different model weights:

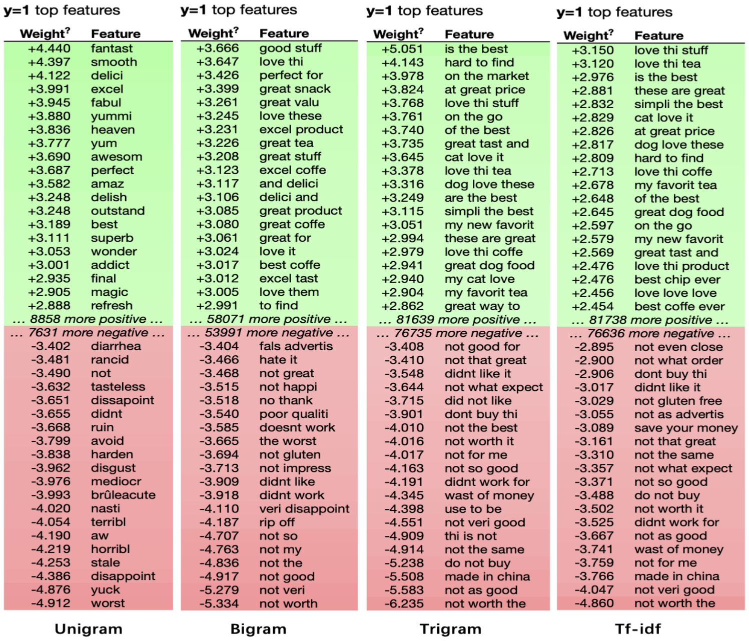
According to different sizes of bag of words and tf-idf, the weighted words look quite different but the basic positive words are overlapping, for example “good”, “great”, “love”, “best” and so on.

(3) Comparing trigram and tf-idf (both are three-words vector):

The three-words in trigram and tf-idf are almost overlapping but with different weights. So it seems that bags of words and tf-idf can get the similar result based on the same size of word numbers.

(4) Comparing model weights on positive and negative classes:

From unigram, we can see that all high-weighted words are reasonable and comprehensive. However, in bigram, trigram and three word tf-idf, the selected words are not efficient or useful as the ones in unigram. For example, most phrases are made up of "great" and another word in positive classes, while most phrases are combinations including “not” in negative classes. We speculated that these inefficient phrases may be the reason for the low accuracy of bigram, trigram and three word tf-idf models.



y=1 top features

Weight?	Feature
+4.440	fantast
+4.397	smooth
+4.122	delici
+3.991	excel
+3.945	fabul
+3.880	yummi
+3.836	heaven
+3.777	yum
+3.690	awesom
+3.687	perfect
+3.582	amaz
+3.248	delish
+3.248	outstand
+3.189	best
+3.111	superb
+3.053	wonder
+3.001	addict
+2.935	final
+2.905	magic
+2.888	refresh
… 8858 more positive …	
… 7631 more negative …	
-3.402	diarrhea
-3.481	rancid
-3.490	not
-3.632	tasteless
-3.651	dissapoint
-3.655	didnt
-3.668	ruin
-3.799	avoid
-3.838	harden
-3.962	disgust
-3.976	mediocr
-3.993	brûleacute
-4.020	nasti
-4.054	terribl
-4.190	aw
-4.219	horribl
-4.253	stale
-4.386	disappoint
-4.876	yuck
-4.912	worst

Unigram

y=1 top features

Weight?	Feature
+3.666	good stuff
+3.647	love thi
+3.426	perfect for
+3.399	great snack
+3.261	great valu
+3.245	love these
+3.231	excel product
+3.226	great tea
+3.208	great stuff
+3.123	excel coffe
+3.117	and delici
+3.106	delici and
+3.085	great product
+3.080	great coffe
+3.061	great for
+3.024	love it
+3.017	best coffe
+3.012	excel tast
+3.005	love them
+2.991	to find
… 58071 more positive …	
… 53991 more negative …	
-3.404	fals advertis
-3.466	hate it
-3.468	not great
-3.515	not happi
-3.518	no thank
-3.540	poor qualiti
-3.585	doesnt work
-3.665	the worst
-3.694	not gluten
-3.713	not impress
-3.909	didnt like
-3.918	didnt work
-4.110	veri disappoint
-4.187	rip off
-4.707	not so
-4.763	not my
-4.836	not the
-4.917	not good
-5.279	not veri
-5.334	not worth

Bigram

y=1 top features

Weight?	Feature
+5.051	is the best
+4.143	hard to find
+3.978	on the market
+3.824	at great price
+3.768	love thi stuff
+3.761	on the go
+3.740	of the best
+3.735	great tast and
+3.645	cat love it
+3.378	love thi tea
+3.316	dog love these
+3.249	are the best
+3.115	simpli the best
+3.051	my new favorit
+2.994	these are great
+2.979	love thi coffe
+2.941	great dog food
+2.940	my cat love
+2.904	my favorit tea
+2.862	great way to
… 81639 more positive …	
… 76735 more negative …	
-3.408	not good for
-3.410	not that great
-3.548	didnt like it
-3.644	not what expect
-3.715	did not like
-3.901	dont buy thi
-4.010	not the best
-4.016	not worth it
-4.017	not for me
-4.163	not so good
-4.191	didnt work for
-4.345	wast of money
-4.398	use to be
-4.551	not veri good
-4.909	thi is not
-4.914	not the same
-5.238	do not buy
-5.508	made in china
-5.583	not as good
-6.235	not worth the

Trigram

y=1 top features

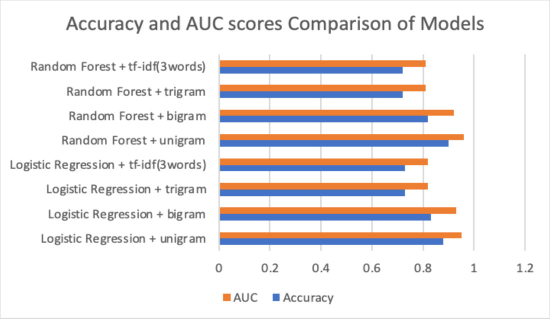
Weight?	Feature
+3.150	love thi stuff
+3.120	love thi tea
+2.976	is the best
+2.881	these are great
+2.832	simpli the best
+2.829	cat love it
+2.826	at great price
+2.817	dog love these
+2.809	hard to find
+2.713	love thi coffe
+2.678	my favorit tea
+2.648	of the best
+2.645	great dog food
+2.597	on the go
+2.579	my new favorit
+2.569	great tast and
+2.476	love thi product
+2.476	best chip ever
+2.456	love love love
+2.454	best coffe ever
… 81738 more positive …	
… 76636 more negative …	
-2.895	not even close
-2.900	not what order
-2.906	dont buy thi
-3.017	didnt like it
-3.029	not gluten free
-3.055	not as advertis
-3.089	save your money
-3.161	not that great
-3.310	not the same
-3.357	not what expect
-3.371	not so good
-3.488	do not buy
-3.502	not worth it
-3.525	didnt work for
-3.667	not as good
-3.741	wast of money
-3.759	not for me
-3.766	made in china
-4.047	not veri good
-4.860	not worth the

Tf-idf

Figure4: Weights of Logistic Regression

Comparison between logistic regression model and random forest model

We created logistic regression and random forest models for four different kinds of vectors separately. Then,we compare the AUC and accuracy scores of them. As shown in Figure 5 and Figure 6, the Random Forest + unigram model got the highest score.



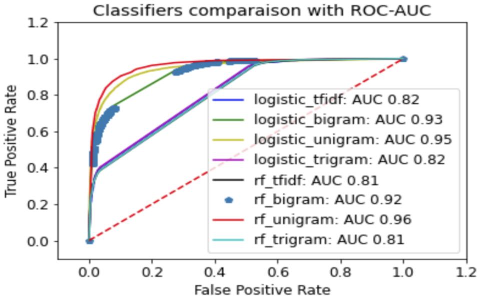


Figure5:Comparison of Accuracy and AUC of Models

Figure6:AUC Comparison between Classifiers

After comparing the metrics performance, we found some interesting conclusions:

(1) Comparing four different kinds of vectors:

To our surprise, unigram always works the best in two different kinds of models. At the beginning, we thought that a vector with three words can contain more information, which means more specific summary and fewer interruptions. But after reviewing the chosen frequent words in model weights, we found the reason: more words does not mean more efficiency or more generalization. But if we continue trying tf-idf with one word, there may be different outcomes.

(2) Comparing two kinds of models:

As we can see, under the best performance vector – unigram, the random forest classifier works the best, with the accuracy of 90%. We also visualized the four different tree plots, and we found that the random forest tree with a unigram got the most complex plots (as shown in Figure 7), which is also the most accurate one.

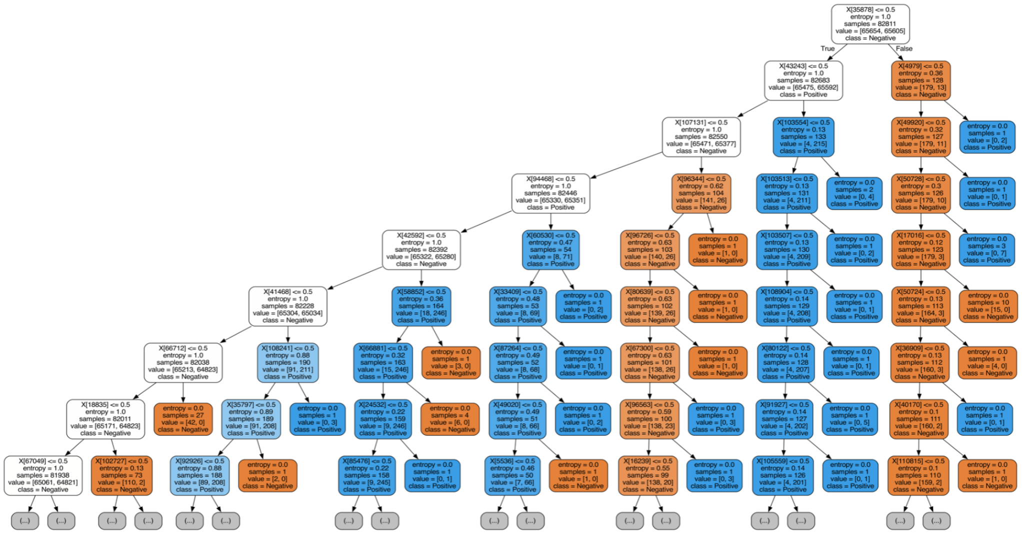


Figure7:Tree Diagram of Models

7. Conclusions and Limitations:

Based on our analysis and observations, among all the models we used, the unigram-randomforest is the better one as a NLP model for the Amazon Food Reviews case. However, due to the limited nature of the hardware and the large size of the dataset, we couldn't run cross validation to find the best parameters for the models successfully. Otherwise, we may get better performance.

In the following exploration, we can compare more kinds of different models, like SVM, Adaboost, XGBoost, Gradient Boosting and so on. For future studies, we would definitely use other machine learning techniques to examine the dataset. The more types of models we run, the better the validation and reliability of our model choice has. Our research will help online shopping platform companies predict customer behavior and response. This expectation will be met after further research using more types of models to determine the best NLP model for Amazon food reviews.

8.References:

1. https://ww.kaggle.com/snap/amazon-fine-food-reviews
2. https://capacity.com/enterprise-ai/faqs/what-are-the-advantages-of-natural-language-processing-nlp/
3. https://medium.com/artefact-engineering-and-data-science/customer-reviews-use-nlp-to-gain-insights-from-your-data-4629519b518e
4. J. McAuley and J. Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. *WWW*, 2013.
5. Acosta Gutiérrez, Gina. (2020). A Comparative Study of NLP and Machine Learning Techniques for Sentiment Analysis and Topic Modeling on Amazon Reviews. 9. 159-170.
6. Et.al, N. K. (2021). Comparative study on Sentiment Analysis Approach for Online Shopping Review. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(3), 1358-1370. doi:10.17762/turcomat.v12i3.907