Amazon Food Reviews Classification for an Intrinsic and Extrinsic Product Quality Assessment

Daniel Elias, B.S. Computer Science Student
Tecnológico de Monterrey, Campus Querétaro, México
A01208905@itesm.mx daniel.eliasbecerra98@gmail.com

Abstract

With the rise of online grocery shopping in platforms such as Amazon, new data sources arise such as the reviews given by customers on the same platform. How can companies use this data to obtain insights that are valuable to each of the different departments within the organization when reviews are varied and evaluate different quality aspects of products? The proposed solution is to classify reviews into two classes: the ones whose contents are related to the intrinsic quality cues of the product, and the ones that are related to the extrinsic aspects. In this report different supervised learning classification algorithms are explored, as well as a deep learning method using a Recurrent Neural Network to perform this Natural Language Processing task.

Introduction

E-commerce platforms for online-grocery shopping have become relevant in recent years. According to Euromonitor International, e-commerce is thriving, registering a 21% growth from 2014 to 2019.[1] Specifically, In the first quarter of 2020, Amazon saw an increase in customer demand in its online grocery category. Amazon's CFO & Senior Vice President, Brian Olsavsky, said the company has tripled grocery sales year after year. [2] This brings about new data provided by customer's product reviews. With these, Data Scientists are now able to uncover new business insights that may drive the decision making process of multinational food manufacturing companies. This report focuses on stating the outcomes of different natural language processing supervised machine learning approaches to classify reviews based on their contents into two categories: intrinsic and extrinsic food quality reviews.

Framing the problem

Amazon food reviews may have more than one sentence for various opinions on different quality aspects of a product. These reviews may be of use to different departments of the manufacturing company; nevertheless, there is data that has no value for other departments. For example, the Food Engineering team may be more concerned in knowing about the consumer's opinion on the taste of a product to measure if a new formula is being accepted by the public; meanwhile, the Marketing department may be more interested in determining the success of an advertising campaign. The problem is: How can the company differentiate different types of reviews based on their contents?

Solution

Defining extrinsic and intrinsic quality cues

According to the European Commission's Joint Research Centre, food quality has a subjective dimension framed by consumer expectations, perceptions, and acceptance. Consumer expectations are based on quality cues that are either intrinsic, which are attributes that are inherent to the product itself and cannot be manipulated without affecting its physical properties (e.g. color, taste, smell), or extrinsic, which are attributes that are not physically a part of a product (e.g. advertising, brand image). [4] Going back to the Food Engineering and Marketing department examples, the first would be more interested in intrinsic-related reviews, while the second on the extrinsic-related opinions. These two categories, intrinsic and extrinsic, are the classes chosen for a binary classification solution. See Table 1 for more examples on both chosen categories.

Food Quality Cues	
Intrinsic	**Extrinsic**
Touch	Packaging
Smell	Price
Taste	Brand image
Shape	Advertising
Color	Mascots
Nutritional value	Delivery
Ingredients	Containers
Chemicals	Servings

Table 1: Food Quality Cues

Finding food reviews

Amazon Food Reviews are used as the data to train the machine learning models. These come from two sources:

- Amazon Fine Food Reviews on Kaggle with 568,454 rows (2012) [6]
- Amazon Grocery and Gourmet Food Reviews with 151,254 rows (2014) [7]

Both data sets contain more than one column, but for this solution only the one containing the review's text is used. It is worth noticing that no data set containing labelling of intrinsic and extrinsic quality was found for the framed problem; thus, requiring a manual labelling that is discussed in the following sections.

Data Preparation

For this project to have a practical example, Kellogg's was the food company selected to analyze their product reviews on Amazon. Reviews from each dataset were filtered in order to use the ones that in their text contained words and phrases related to Kellogg's using regular expressions. The list of text that was looked for is as follows: frosties, kellogg's, pop tart, froot loop, cheez it, corn flake. eggo, nutri-grain, frosted flake, rice krispies, special k, cocoa krispies, tony tiger, toucan sam,pringles, all bran, raisin bran, morningstar farms, corn pop, krave, cheezit.

After only these reviews were extracted, each one of them was separated into sentences using Python's Natural Language Processing Toolkit library (nltk). Then, the text was converted to lowercase and special characters were removed. Finally the data obtained from both datasets was merged into a single csv file.

Exploratory Data Analysis

To ensure the data obtained corresponded to the words and phrases that are of Kellogg's interest, word clouds were created to visualize the most common words in the merged dataset. Image 1 presents one of the resulting visualizations.

Image 1: Kellogg's Amazon Reviews Wordcloud

Labelling

Manual labelling had to be done for the dataset. A target column was added specifying the class for each row of data. A letter 'e' indicates a sentence that mainly has contents regarding the extrinsic quality aspects of the product while a letter 'i' was added when the contents of the sentence mainly evaluated the intrinsic quality. This labelling was carefully done paying attention to the quality cues presented for each category in Table 1.

Predictive Modelling

The proposed solution needs the use of Natural Language Processing techniques to do binary classification of the reviews. Natural Language Processing is a field of Artificial Intelligence that combines computational linguistics with statistical and machine learning models. The aim of it is to make computers understand human language. [28] In order for Machine Learning models to have the ability to use text as input, two tools offered by the scikit-learn library were used: CountVectorizer and TfidfVectorizer. These both approaches convert each word or ngram of a sentence into a feature, the difference is that CountVectorizer only counts the number of appearances of each word or ngram in each review, while Term Frequency - Inverse Document Frequency Vectorizer, better known as TF-IDF, Measure of originality of a word by comparing the times a word appears in a document with number of documents the word appears in. In relation to this solution, the documents are the reviews. [17] TF-IDF is calculated using the following formula [29] :

For a term i in document j:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

To find the best solution to the problem, both CountVectorizer and TfidfVectorizer were used in combination with eight different Machine Learning algorithms offered by the scikit-learn library: Linear Support Vector Machine, Logistic Regression, Multinomial Naive Bayes, Random Forest, Linear SVM Stochastic Gradient Descent, Bagging Random Forest, Gradient Boosting and AdaBoost. These algorithms were chosen based on their ability to perform binary classification.

The top five best performing models regarding their test accuracy appear on Table 2.1. The complete table with the accuracy scores for the sixteen models can be found in the appendix in Table 2.2. A graphical comparison of the train and test accuracy scores is displayed in Chart 1.1. The complete chart with the test vs. train accuracy scores for the sixteen models can be found in the appendix in Chart 1.2

Feature Vector	Classifier	Train Accuracy	Test Accuracy
Tfidf Vect	Linear SVM	0.9880	0.7857
Count Vect	Logistic Regression	0.9817	0.7714
Count Vect	Multinomial Naive Bayes	0.9173	0.7714
Count Vect	Random Forest	0.9992	0.7523
Tfidf Vect	Stochastic Gradient Descent	0.9960	0.7523

Table 2.1: Best test accuracy scores

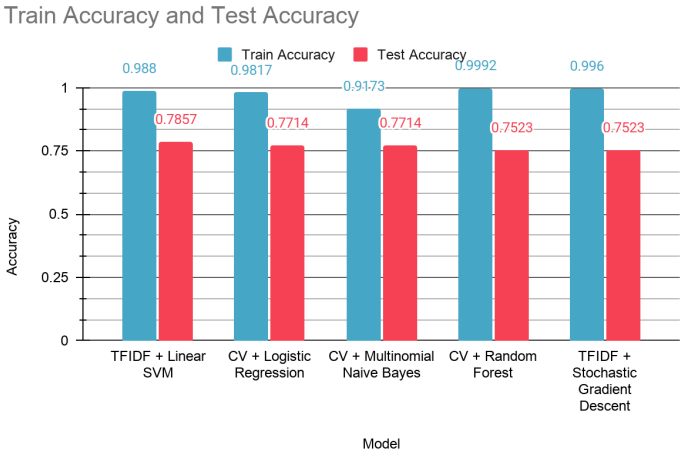

Chart 1.1: Grouped bar chart of the best test accuracy scores

Hyperparameter Tuning

After the last step, hyperparameter tuning was performed upon the top five best performing algorithms mentioned earlier alongside two other models were selected to do hyperparameter tuning. This was made using scikit-learn RandomizedSearchCV class to find the best hyperparameters for each algorithm in combination with the TF-IDF and Count vectorizers. The results for each model after being tuned can be seen on Chart 2.1. A comparison of the test accuracy before and after tuning can be seen on Chart 2.2.

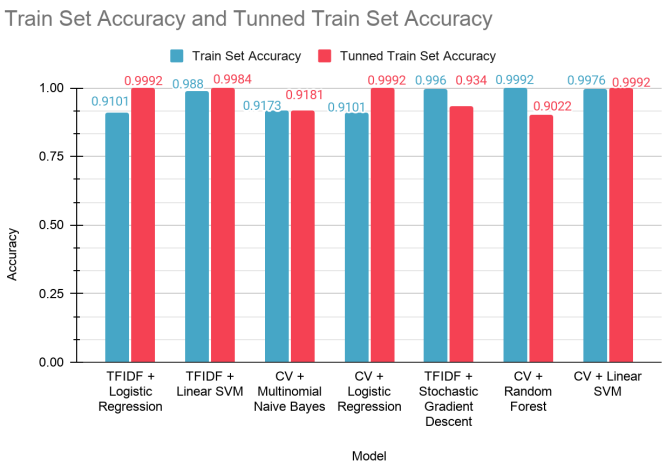

Chart 2.1: Train and test accuracy scores of selected tuned models

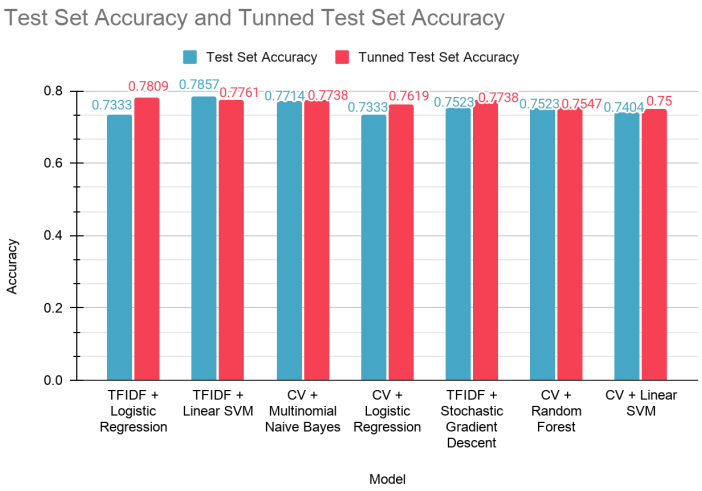

Chart 2.2: Test accuracy scores of models before and after tuning

From Chart 2.2. Is worth noticing that the test accuracy of the TF-IDF + Linear SVM decreased after tuning. On the other hand, the test accuracy of TF-IDF + Logistic Regression augmented considerably from 0.7333 to 0.7809 positioning it as the second best model. A complete table of this comparison can be found on Table 3 in the appendix.

Recurrent Neural Network

To explore more Natural Language Processing approaches, a recurrent neural Network using the Keras library was included. A recurrent neural network (RNN) is a deep learning algorithm commonly used in NLP tasks. Thanks to its internal memory, these neural networks can remember the input they previously received, making them very precise in predicting what is coming next. [30] For example, in a sentence by remembering the first words the next word is more easily predicted. Chart 3.1 shows the train and test accuracy of the implemented RNN. Chart 3.2 is its confusion matrix.

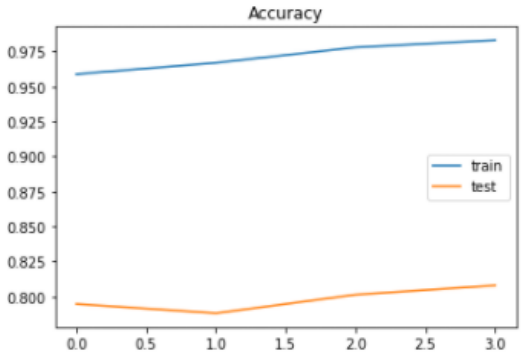

Chart 3.1 : Train and test accuracy of the RNN

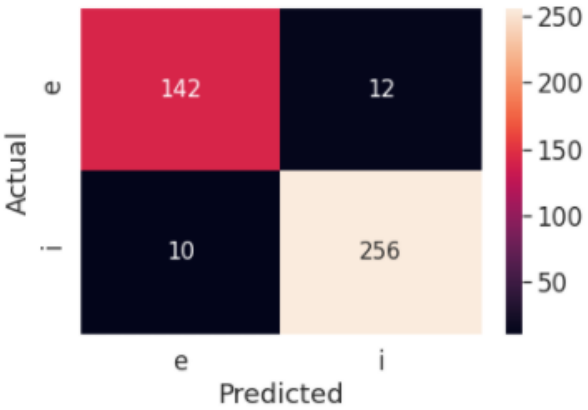

Chart 3.2 : Confusion matrix of the RNN

The implementation of the RNN using Keras obtained 80% test accuracy, positioning it as the best model to solve the problem. Another representation of these results is shown on Chart 3.2, having 398 correctly predicted samples and only 22 predicted incorrectly.

Results

Based on the test accuracy score of the explored models the six selected models to perform the binary classification are:

1. Recurrent Neural Network using Keras
2. TF-IDF and Linear SVM
3. TF-IDF and Logistic Regression
4. Count Vectorizer and Multinomial Naive Bayes
5. TF-IDF and Stochastic Gradient Descent Linear SVM
6. Count Vectorizer and Random Forest

With the highest test accuracy score being 80% by the Recurrent Neural Network using Keras

Future work

Future exploration on the data preparation step can be included by exploring the change in accuracy when adding lemmatization, stop word removal and spelling correction. Moreover, an unsupervised learning clustering algorithm could be added to perform the labelling of the reviews.

Conclusion

The proposed solution explored different approaches to performing the Natural Language Processing task of classifying text into two classes: extrinsic and intrinsic quality. These can be further exploited by companies to only use data valuable to them and certain departments, reducing their need for storage space and time invested in analyzing irrelevant data for a particular team within the organization.

Appendix

Table 2.2: Comparison of 16 approaches for Natural Language Binary Classification

Feature Vector Type	**Classifier**	**Train Set Accuracy**	**Test Set Accuracy**
Count Vect	Multinomial Naive Bayes	0.9173	0.7714
Count Vect	Linear SVM	0.9976	0.7404
Count Vect	Logistic Regression	0.9817	0.7714
Count Vect	Stochastic Gradient Descent	0.9976	0.7309
Count Vect	Random Forest	0.9992	0.7523
Count Vect	Bagging Random Forest	0.9817	0.7261
Count Vect	Gradient Boosting	0.8616	0.7404
Count Vect	Ada Boost	0.8163	0.7380
Tfidf Vect	Multinomial Naive Bayes	0.8243	0.6928
Tfidf Vect	Linear SVM	0.9880	0.7857
Tfidf Vect	Logistic Regression	0.9101	0.7333
Tfidf Vect	Stochastic Gradient Descent	0.9960	0.7523
Tfidf Vect	Random Forest	0.9992	0.7214
Tfidf Vect	Bagging Random Forest	0.9809	0.7095
Tfidf Vect	Gradient Boosting	0.8895	0.7285
Tfidf Vect	Ada Boost	0.8179	0.7119

Chart 1.2: Grouped bar chart of 16 approaches for Natural Language Binary Classification

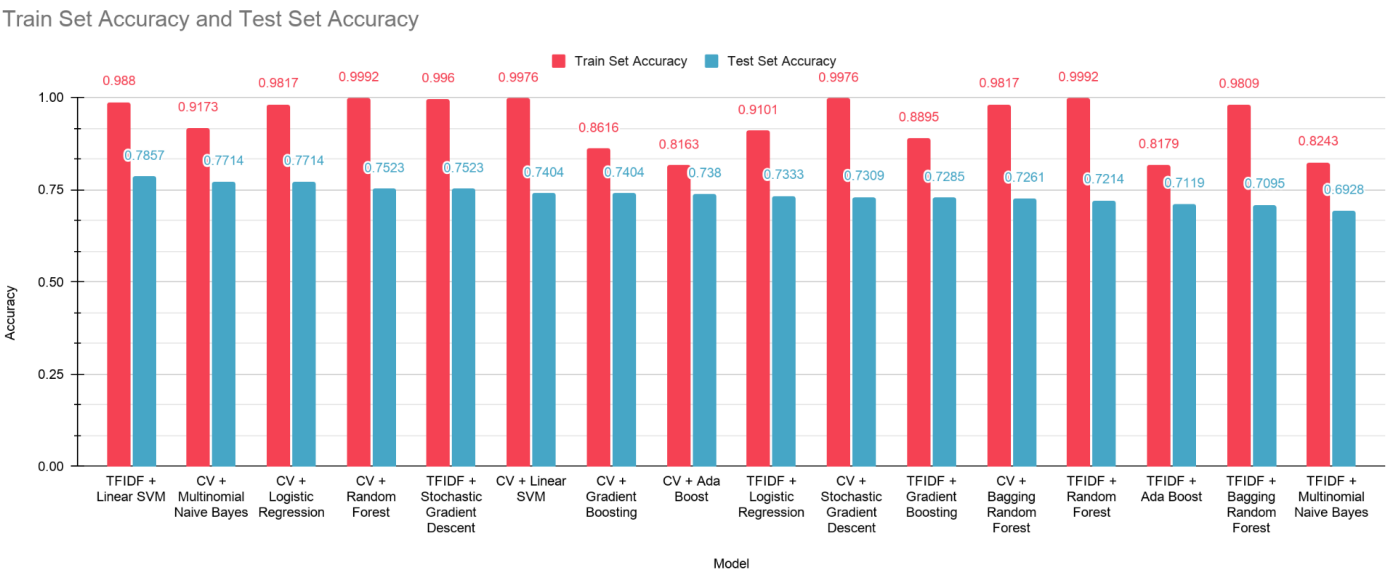

Table 3: Comparison of 7 approaches for Natural Language Binary Classification after Hyperparameter Tuning

Feature Vector Type	**Classifier**	**Train Set Accuracy**	**Test Set Accuracy**
Count Vect	Multinomial Naive Bayes	0.9181	0.7738
Count Vect	Linear SVM	0.9992	0.75
Count Vect	Logistic Regression	0.9992	0.7619
Count Vect	Random Forest	0.9022	0.7547
Tfidf Vect	Linear SVM	0.9984	0.7761
Tfidf Vect	Logistic Regression	0.9992	0.7809
Tfidf Vect	Stochastic Gradient Descent	0.9920	0.7619

References

[1]https://blog.euromonitor.com/covid-19-to-accelerate-online-grocery-shopping-beyond-2021/
[2]https://www.supermarketnews.com/online-retail/amazon-online-grocery-sales-triple-second-quarter
[3]https://www.mdpi.com/2071-1050/12/22/9594/pdf
[4]https://ec.europa.eu/jrc/sites/jrcsh/files/eu_harmonised_testing_methodology_-_framework_for_selecting_and_testing_of_food_products_to_assess_quality_related_characteristics.pdf
[5]https://www.mdpi.com/2304-8158/9/4/396/pdf
[6]https://www.kaggle.com/snap/amazon-fine-food-reviews
[7]http://jmcauley.ucsd.edu/data/amazon/index_2014.html
[8]http://deepyeti.ucsd.edu/jianmo/amazon/index.html
[9]https://colab.research.google.com/drive/1Zv6MARGQcrBbLHyjPVVMZVnRWsRnVMpV
[10]https://relatedwords.org/relatedto/kellogg's
[11]https://dataconomy.com/2016/10/big-data-python/
[12]https://ch-nabarun.medium.com/read-json-using-pyspark-f792bda95741
[13]geeksforgeeks.org/python-lemmatization-with-nltk/
[14]nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html
[15]scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
[16]scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
[17]medium.com/@cmukesh8688/tf-idf-vectorizer-scikit-learn-dbc0244a911a
[18]scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html
[19]scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
[20]scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
[21]scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
[22]scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html
[23]scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html
[24]towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989
[25]scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
[26]scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html
[27]scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html
[28]https://www.ibm.com/cloud/learn/natural-language-processing
[29]https://medium.com/analytics-vidhya/an-introduction-to-tf-idf-using-python-5f9d1a343f77
[30] https://www.ibm.com/cloud/learn/recurrent-neural-networks