Amazon Fine Food Reviews Data Analysis Project using Machine Learning Techniques

By

Ann-Marie Mensah

Abstract

Every day, millions of insightful product and service reviews are generated on the internet and it has become imperative for businesses to use these reviews to provide excellent services for their customers. Not only do these reviews help companies achieve that, but they also enable other customers make the right decisions before buying any product or using any service. Analysing large amounts of reviews can be an overwhelming process for product manufacturers. This project considers this process a classification problem, that is, classifying opinions into positive or negative reviews.

Three machine learning classifiers, K Nearest Neighbors, Logistic Regression and Naïve Bayes, were used to analyze fine food reviews from the Amazon Fine Food Reviews dataset and make predictions as well. All three classifiers performed well however, the results showed that, in terms of accuracy, Logistic Regression achieves better prediction results than Naïve Bayes and K Nearest Neighbors.

Section 1

Introduction

In today's hypercompetitive business environment, more firms in the Fast-Moving Consumer Goods (FMCG) industry are relying heavily on customer reviews to meet the needs of their customers. According to Jiang et al. (2017), online customer reviews for products and services have become extremely popular because they contain rich information about customer expectations. Essentially, these reviews can help firms create a unique experience for their customers (Rahman, 2014).

Amazon Inc., which was originally an online marketplace for books founded by Jeff Bezos, is now an American multinational technology company which focuses on e-commerce, cloud computing, digital streaming, and artificial intelligence (Wikipedia, 2022). Customer reviews, according to Mudambi and Schuff (2010), are peer-generated evaluations of products posted on company or third-party websites, and Amazon has enabled this feature for its customers for many years. Customer reviews have shown to have a positive effect on sales, attract customer visits and show a sense of community for online shoppers (Mudambi & Schuff, 2010).

In recent years, machine learning methods, like sentiment analysis, have helped companies to analyse how their customers feel about their products and/or services and to find ways to meet their needs better. As part of that conversation, this project considered a classification analysis for Amazon fine food reviews from the Amazon Fine Food Reviews dataset on Kaggle.

Problem Statement

Every day, millions of insightful product and service reviews are generated on the internet and it has become imperative for businesses to use these reviews to provide excellent services for their customers. Not only do these reviews help companies achieve that, but it also

enables other customers make the right decisions before buying any product or using any service. Analysing large amounts of reviews can be an overwhelming and time-consuming process for product manufacturers. To better understand how customers feel about products and services, companies need to adopt the most suitable machine learning techniques to determine the overall semantics of their customers' reviews.

In lieu of that, this sentiment analysis project aims to answer this question:

Which machine learning approach is better performing, in terms of accuracy, on Amazon Fine Food Reviews: Logistic Regression, K-Nearest Neighbors or Naïve Bayes?

Goals and Objectives

The goals and objectives for this sentiment analysis project are:

1. To evaluate which machine learning technique(s) is/are the best for analysing Amazon Fine Food Reviews
2. To make predictions for Amazon Fine Food Reviews

Organization of Sentiment Analysis Project

The rest of this project is as follows:

Section 2, which is related work, looks at literature reviews of sentiment analysis experiments. In section 3, which is techniques and methodology, the type of analysis that was performed, the classifiers that were used in the sentiment analysis, the process of collecting and pre-processing data, and how the machine learning classifiers were applied on the dataset was discussed. Discussion of findings was gathered in section 4 and section 5 concluded the study.

Section 2

Related Work

Mudambi and Schuff (2010), in their paper, analysed the helpfulness of online reviews using a Tobit regression. The Tobit regression was the best model they used because the dependent variable (helpfulness) was limited at the extremes. This means that customers can either vote for a review to be helpful or not, and nothing more. This study focused on six consumer products thus, the results were focused on those products only. Further studies could sample more products to make the results more generalized. The insights that emerged from the study were that product type improves the understanding of reviews to customers and a lengthier review increases the helpfulness of the said reviews.

Baid, Chaplot and Gupta (2017), in their paper, performed a sentiment analysis on movie reviews using Naïve Bayes, K-Nearest Neighbour and Random Forest. The sentiment analysis was conducted on the WEKA tool, and it was concluded that for movie reviews, the Naïve Bayes classifier is the best method for analysis as it showed an accuracy score of 81.45%. The Random Forest classifier achieved a 78.65% accuracy score, and the K-Nearest Neighbour classifier achieved a 55.30% accuracy score.

Deshmukh, Jagdale, Shirsat (2019), in their paper, analyzed reviews of the following products from Amazon: cameras, laptops, mobile phones, TVs and video surveillance. The product reviews were classified based on positive and negative opinions. The paper concluded with Naïve Bayes achieving an accuracy score of 98.17% and the Support Vector Machine achieving an accuracy score of 93.54% for Camera Reviews.

Coyne, Güner, and Smit (2019), in their paper, investigated if product reviews were feasible for Amazon products. In the study, Multinomial Naive Bayes (MNB), Linear Support Vector Machine (LSVM) and Long Short-term Memory Network (LSTM) were the machine learning algorithms that were compared, trained, and tested on an Amazon dataset with 4

million reviews from Kaggle. The LSTM had the best accuracy score of 0.90. The LTSM was applied on the remaining reviews from the dataset, as well as on a new scraped dataset from Amazon.com containing product reviews from different categories. This resulted in a very accurate classification, with furniture products achieving an accuracy score of 0.92. In conclusion, LSTM networks are very suitable for binary sentiment analysis on Amazon product reviews and the results do not change significantly for different categories (Coyne, Güner, and Smit, 2019).

Section 3

Techniques and Methodology

This section discusses the type of analysis that was performed, the classifiers that were used in the sentiment analysis, the process of collecting and pre-processing data, and how the machine learning classifiers were applied on the dataset.

Sentiment Analysis

Millions of reviews are generated by customers every day thus, it can overwhelmingly be difficult for companies to keep track of the opinions of their customers. According to Paknejad (2018), "classification is a way to tackle that problem. It is a computational study which uses subjective information from the given texts in natural language, such as opinions and sentiments." In recent years, machine learning methods have become popular to use in sentiment analysis for their simplicity and accuracy.

Sentiment analysis, a natural language processing method, is the process of discovering patterns and extracting information from textual documents and resources. The main goal of sentiment analysis is to classify textual documents into positive and negative opinions. According to Baid, Chaplot and Gupta (2017), sentiment analysis has been used in many fields such as marketing and decision making, quality improvement, as well as building recommendation systems.

K Nearest Neighbors

K-nearest neighbors is one of the simplest algorithms in machine learning. "The principle behind this method is to find a predefined number of training samples closest in distance to the new point and predict the label from these" (Baid, Chaplot and Gupta, 2017).

The algorithm looks for similar texts in the set of training texts so that if the texts have *n* attributes, then it will consider them as a vector in n-dimensional space and predict the class

label of the new text based on a distance criterion in this space such as the Euclidean distance as well as the class label of the neighbours (Kuhkan, 2016).

Naïve Bayes

The Naïve Bayes classifier assumes that features (which are usually words) in a class are independent from any other features. This algorithm is useful for very large datasets (Baid, Chaplot and Gupta, 2017). According to Rish (2001), "Naïve Bayes has proven to be effective in many practical applications, including text classification, medical diagnosis, and systems performance management".

Logistic Regression

Logistic Regression is one of the most common methods for solving binary classification problems. It predicts binary classes and computes the probability of an event occurring (Anu, Gladence and Karthi, 2015).

Data Collection

In this sentiment analysis project, data was captured from the Amazon's Fine Food Reviews dataset on Kaggle. The dataset, consisting of 74,258 different products, was collected between October 1999 and October 2012 with a total of 568,454 customer reviews.

Fig 1. **Amazon Fine Food Reviews dataset**

```
reviews = pd.read_csv('amazonreviews.csv', quoting=3, on_bad_lines='skip')
reviews.head()
```

```
/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py:3326: DtypeWarning: Columns (4,5) have mix
  exec(code_obj, self.user_global_ns, self.user_ns)
```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5.0	1.303862e+09
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1.0	1.346976e+09
2	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2.0	1.307923e+09

Each review was included under the following ten (10) features:

Table 1. **Description of Features in the Amazon Fine Food Reviews dataset**

Feature	**Description**
Id	Row Id
ProductId	Unique identifier of the product
UserId	Unique identifier of the user
ProfileName	Profile name of the user
HelpfulnessNumerator	Number of users who found the review helpful
HelpfulnessDenominator	Number of users who indicated whether they found the review helpful or not
Score	Rating between 1 and 5
Time	Timestamp of the review
Summary	Summary of the review
Text	Full text of the review

Data Pre-processing

Dropping Unnecessary Columns

For data pre-processing, the following columns were dropped since they were not needed in the analysis: *ProductId, Time, HelpfulnessNumerator, HepfulnessDenominator* and *ProfileName.* The remaining features were Id, UserId, Score, Summary and Text, as seen in Fig 2 below.

Fig 2. **Dropping of Unnecessary Columns**

```
#Dropping labels

columns = ['Id', 'ProductId', 'UserId', 'ProfileName', 'HelpfulnessNumerator',
           'HelpfulnessDenominator', 'Score', 'Time', 'Summary', 'Text']
new_reviews = reviews.drop(columns=['ProfileName','Time', 'HelpfulnessNumerator',
                                    'HelpfulnessDenominator'], axis=1)
new_reviews.head()
```

	Id	ProductId	UserId	Score	Summary	Text
0	1	B001E4KFG0	A3SGXH7AUHU8GW	5.0	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	B00813GRG4	A1D87F6ZCVE5NK	1.0	Not as Advertised	"Product arrived labeled as Jumbo Salted Peanu...
2	4	B000UA0QIQ	A395BORC6FGVXV	2.0	Cough Medicine	If you are looking for the secret ingredient i...
3	9	B000E7L2R4	A1MZYO9TZK0BBI	5.0	Yay Barley	Right now I'm mostly just sprouting this so my...
4	10	B00171APVA	A21BT40VZCCYT4	5.0	Healthy Dog Food	This is a very healthy dog food. Good for thei...

Tokenizing and Removal of Stopwords

Tokenizing is the process of breaking down texts or sentences into smaller words, or other smaller units called tokens. After breaking down the 'Summary' texts into tokens, I filtered out the stopwords, as seen in Fig 3 below. Stopwords, such as articles, prepositions, and conjunctions, are words that are of little value to machine learning models. Removing them can enhance the performance of the machine learning models. Examples are "a", "why", and "the".

Fig 3. **Tokenizing and Removal of Stopwords**

```
# Removing stopwords

letters_only = re.sub("[^a-zA-Z]",  # Search for all non-letters
                      " ",          # Replace all non-letters with spaces
                      str(new_reviews['Summary']))
tokens = word_tokenize(letters_only)
filtered_tokens = [word for word in tokens if not word in stopwords.words('english')]
filtered_tokens
```

```
['Good',
 'Quality',
 'Dog',
 'Food',
 'Not',
 'Advertised',
 'Cough',
 'Medicine',
 'Yay',
 'Barley',
 'Healthy',
 'Dog',
 'Food',
 'Wonderful',
```

Creating Positive and Negative Reviews Score Column

To make the score ratings of Amazon Fine Food Reviews simpler (Score column), all scores below 3 were made negative reviews, with a new score label of 0, and all scores above 3 were made positive reviews, with a new score label of 1, in a new column called 'positive_negative'. All neutral views (reviews with a score of 3) were eliminated, as seen in Fig 4 below.

Fig 4. **Creating Positive and Negative Reviews Score Column and Removing Neutral Reviews**

```
[11] #Removing Neutral Views

     new_reviews['Score'] = pd.to_numeric(new_reviews['Score'], errors='coerce')
     new_reviews.dropna(inplace=True)
     new_reviews[new_reviews['Score'] != 3]
     new_reviews['positive_negative'] = np.where(new_reviews['Score'] > 3, 1, 0)
     new_reviews.head()
```

	Id	ProductId	UserId	Score	Summary	Text	positive_negative
0	1	B001E4KFG0	A3SGXH7AUHU8GW	5.0	Good Quality Dog Food	I have bought several of the Vitality canned d...	1
1	2	B00813GRG4	A1D87F6ZCVE5NK	1.0	Not as Advertised	"Product arrived labeled as Jumbo Salted Peanu...	0
2	4	B000UA0QIQ	A395BORC6FGVXV	2.0	Cough Medicine	If you are looking for the secret ingredient i...	0
3	9	B000E7L2R4	A1MZYO9TZK0BBI	5.0	Yay Barley	Right now I'm mostly just sprouting this so my...	1
4	10	B00171APVA	A21BT40VZCCYT4	5.0	Healthy Dog Food	This is a very healthy dog food. Good for thei...	1

Balancing the Positive-Negative Reviews Ratio

An imbalanced classification problem is a classification problem where the distribution is biased or skewed. Brownlee (2019) argues that imbalanced classification problems pose a challenge for predictive modelling as most of the machine learning algorithms used for classification problems were designed around the assumption of an equal number of examples for each class. This results in models having poor predictive performance, specifically for the minority class. This is a problem because the minority class is more important and therefore more sensitive to classification errors than the majority class (Brownlee, 2019).

From the dataset, the positive reviews had the highest distribution as compared to the negative reviews, causing an imbalance in the entire dataset. Since the negative reviews were about 60,000 in total, the 'positive-negative' reviews cap was set at 13,000, as seen in Fig 5 below. A bar chart comparison of the imbalanced and balanced positive-negative reviews can be seen in Fig 6. The dataset had to be balanced because the classification report for the minority class, which is the negative reviews, would have performed poorly with the imbalanced data.

Fig 5. Balancing the Positive-Negative Reviews Ratio

Balancing positive_negative Score Distribution

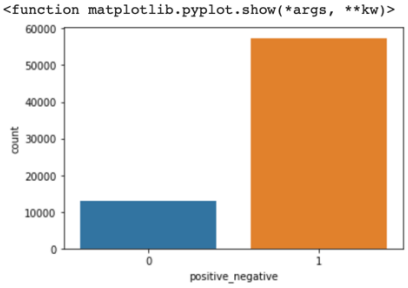
```
[21] ### Balancing positive_negative Score Distribution

     pos = new_reviews[new_reviews['positive_negative']==1]
     neg = new_reviews[new_reviews['positive_negative']==0]
     new = pd.concat([pos[0:13000], neg[0:13000]], axis=0)
     new.head()
```

	Id	ProductId	UserId	Score	Summary	Text	positive_negative
0	1	B001E4KFG0	A3SGXH7AUHU8GW	5.0	Good Quality Dog Food	I have bought several of the Vitality canned d...	1
3	9	B000E7L2R4	A1MZYO9TZK0BBI	5.0	Yay Barley	Right now I'm mostly just sprouting this so my...	1
4	10	B00171APVA	A21BT40VZCCYT4	5.0	Healthy Dog Food	This is a very healthy dog food. Good for thei...	1
5	14	B001GVISJM	A18ECVX2RJ7HUE	4.0	fresh and greasy!	good flavor! these came securely packed... the...	1
6	15	B001GVISJM	A2MUGFV2TDQ47K	5.0	Strawberry Twizzlers - Yummy	The Strawberry Twizzlers are my guilty pleasur...	1

Fig 6. **Bar Chart of Imbalanced and Balanced Positive-Negative Reviews**

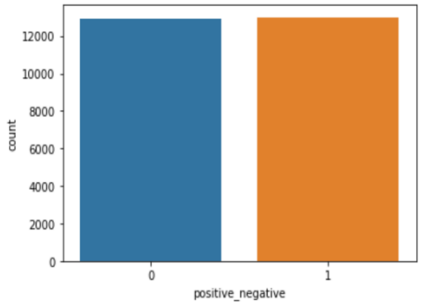
```
#Showing Positive and Negative Score Distribution

sns.countplot(new_reviews['positive_negative'])
plt.show
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py
  FutureWarning
<function matplotlib.pyplot.show(*args, **kw)>
```




```
[22] sns.countplot(new['positive_negative'])
     plt.show
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py
  FutureWarning
<function matplotlib.pyplot.show(*args, **kw)>
```



Section 4

Discussion of Findings

Splitting Data into Train and Test

Three classifiers were used in this project: K Nearest Neighbors, Naïve Bayes, and Logistic Regression. Each classifier was trained and then tested. The classifiers were trained and tested on the reviews scores and on the summary of the reviews. The 80-20 train-test split was used for this project. A corpus of 52,734 data was collected as training data set and the remaining 17,579 for testing the accuracy of the classifiers, as seen in Fig 7 below.

Fig 7. **Splitting Data into Train and Test**

```
x = new_reviews['Summary'] #input
y = new_reviews['positive_negative'] #output
x_train, x_test, y_train, y_test = train_test_split(x, y)

print("The size of x_train:", x_train.shape)
print("The size of y_train:", y_train.shape)
print("The size of x_test:", x_test.shape)
print("The size of y_test:", y_test.shape)
```

```
The size of x_train: (52734,)
The size of y_train: (52734,)
The size of x_test: (17579,)
The size of y_test: (17579,)
```

Feature Engineering

Machine learning algorithms deal with numbers, so texts need to be transformed into numbers before feeding them to the algorithms. This process was done by using the TF-IDF Vectorizer. TF-IDF associates each word with a number so the models can recognize the data. The texts were then fitted and transformed and stored in new variables, as seen in Fig 8 below.

Fig 8. **TF-IDF Vectorizer**

```
[24] vectorizer = TfidfVectorizer(min_df=5)
     vector = vectorizer.fit(x_train)

     new_x_train = vectorizer.transform(x_train).toarray()
     new_x_test = vectorizer.transform(x_test).toarray()

     print("The size of new_x_train:", new_x_train.shape)
     print("The size of new_x_test:", new_x_test.shape)
```

```
The size of new_x_train: (52734, 2795)
The size of new_x_test: (17579, 2795)
```

Predicting Sentiments

The three classifiers were then trained and tested on the new train and test sets. I also predicted sentiments using all three classifiers, and they all predicted correctly, as seen in Fig 9 below.

Fig 9. **Predicting Sentiments**

```
#Logistic Regression

statements = ["The food is tasty", "The food is horrible"]
encoded_statements = vectorizer.transform(statements).toarray()
predicted_class = model_1.predict(encoded_statements)
predicted_class
```

```
array([1, 0])
```

```
[32] #Naive Bayes

statements = ["The food is tasty", "The food is horrible"]
encoded_statements = vectorizer.transform(statements).toarray()
predicted_class = model_2.predict(encoded_statements)
predicted_class
```

```
array([1, 0])
```

```
[33] #K Nearest Neighbors

statements = ["The food is tasty", "The food is horrible"]
encoded_statements = vectorizer.transform(statements).toarray()
predicted_class = model_3.predict(encoded_statements)
predicted_class
```

```
array([1, 0])
```

Accuracy

For accuracy, the Logistic Regression classifier had a score of 0.90, followed by Naïve Bayes and K Nearest Neighbors with 0.89. These results mean that, in terms of prediction accuracy, Logistic Regression achieves better results than Naïve Bayes and K Nearest Neighbors for Amazon Fine Food Reviews.

Section 5

Limitations of Study

Only TF-IDF was used to represent texts in this sentiment analysis project. Further studies can be done where Bag of words or Bag of n-grams are used, and results compared.

Also, since just three machine learning classifiers were examined in this sentiment analysis project, future research can be done to explore other classifiers like Decision Tree and Support Vector Machines.

Section 6

Conclusion

I examined the Amazon Fine Food Reviews dataset for this project. A literature review was conducted to get familiar with the works that had been done in the text classification domain. It was followed by a step-by-step write up on data collection, text pre-processing and application of machine learning classifiers. Finally, the classifiers were trained and tested and evaluated. All three classifiers performed well however, the results showed that, in terms of accuracy, Logistic Regression achieves better results than Naïve Bayes and K-NN in the analysis of Amazon Fine Food Reviews. The Logistic Regression model had the best accuracy score of 90%, making it the best model in predicting Amazon Fine Food Reviews even though K-NN and Naive Bayes performed well too with an accuracy score of 89%.

References

Anu, V.M., Gladence, L.M., Karthi, M. (2015). A Statistical Comparison of Logistic Regression and Different Bayes Classification Methods for Machine Learning. *ARPN Journal of Engineering and Applied Sciences*, *10*(14), pp.5947-5953.

Baid, P., Chaplot, N., and Gupta, A. (2017). Sentiment Analysis of Movie Reviews Using Machine Learning Techniques. *International Journal of Computer Applications*, *179*(7), pp.45-49.

Brownlee, J. (2019). A Gentle Introduction to Imbalanced Classification. *Machine Learning Mastery*, *22*.

Coyne, E., Güner, L., and Smit, J. (2019). Sentiment Analysis for Amazon.com Reviews. *Big Data in Media Technology (DM2583) KTH Royal Institute of Technology, Stockholm*.

Deshmukh, S.N., Jagdale, R.S., Shirsat, V.S. (2019). Sentiment Analysis on Product Reviews Using Machine Learning Techniques. In *Cognitive Informatics and Soft Computing* (pp. 639-647). Springer, Singapore.

Jiang, H., Kwong, C.K. and Yung, K.L. (2017). Predicting Future Importance of Product Features Based on Online Customer Reviews. Journal of Mechanical Design, 139(11), p.111413.

Kuhkan, M. (2016). A Method to Improve the Accuracy of K-Nearest Neighbour Algorithm. *International Journal of Computer Engineering and Information Technology*, *8*(6), p.90.

Paknejad, S. (2018). Sentiment Classification on Amazon Reviews Using Machine Learning Approaches. Available at: https://www.diva-portal.org/smash/get/diva2:1241547/FULLTEXT01.pdf [Accessed: 18 January 2022].

Rish, I. (2001). An Empirical Study of the Naive Bayes Classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence* (Vol. 3, No. 22, pp. 41-46).

Wikipedia. (2022). *Amazon (company) - Wikipedia*. Available at: <https://en.wikipedia.org/wiki/Amazon_(company)> [Accessed: 17 January 2022].