Exploring Sentiment Analysis in Amazon Product Reviews

University of Virginia

DS5559: Big Data Analytics

MSDS, Summer 2020

Group 4

Huilin Chang (hc5hq)

Gavien Wiehl (gtw4vx)

Travis Vitello (tjv9qh)

Abstract

Sentiment analysis using Natural Language Processing methods in PySpark was performed on review data taken from Amazon.com. Datasets were derived from Julian McAuley's UCSD data science page and were specifically focused on Amazon's "Movies and TV" and "Baby" product categories. Machine Learning algorithms were employed after processing the Amazon user reviews, including Logistic Regression, Naïve Bayes, and Random Forest, where it was found that the Naïve Bayes approach was superior in classifying user reviews compared to the other methods based on F1 score (85% for "Movies and TV", 88% for "Baby"). Future research would seek to improve these models through adjusting hyperparameters or to explore alternate algorithms, while also considering alternate data sets; however, there is recognition that improving model results may adversely impact computation time for sufficiently large data sets.

Introduction

Julian McAuley, an associate professor at the University of California, San Diego, has available on his UCSD.edu data science page several large datasets including those representing "Amazon Product Data"[1]. Such product Amazon.com data contains reviews partitioned by product categories, such as "Books", "Beauty", "Amazon Instant Video", and more. McAuley's hosted data spans May 1996 through July 2014 and represents not only the Amazon user review text ("**reviewText"**), but also data and metadata including the Amazon product ID ("**asin"**), user's rating of the product on a scale from 1 to 5 ("**overall"**), the date of the review ("**reviewTime"**), among other values. For this study, the group elected to explore how well a user's rating of a product aligns with or can otherwise be predicted based on the text of the corresponding user's review.

The benefit of performing Natural Language Processing (NLP)-based analysis may help inform services like Amazon as to what product categories have offerings which are favorable or unfavorable to consumers so that the product inventory can be adjusted accordingly or user-specific recommendations can be made[2]. Further, the ability to cross-reference a user's text against the user rating of a product can be useful in training models that predict sentiment beyond Amazon's product offerings, particularly in disciplines or platforms whereby a corresponding rating value does not necessarily exist (such as certain social media posts). Such models can be further developed to gain insight into the current cultural *zeitgeist* and for otherwise manifesting trends and patterns useful in areas including commerce, politics, and anthropology[3].

For this study, the Group elected to explore the "Movies and TV" and "Baby" datasets hosted by McAuley. The "Movies and TV" dataset consists of 1,689,188 reviews while "Baby" contains 160,792 reviews. These Amazon product categories were believed by the Group to offer a diverse set of user opinions and reviews, which were expected to be reasonable for applying NLP methods to determine sentiment. To facilitate the processing and analysis of the data, the Group leveraged PySpark with some small reliance on Python where applicable. For detailed methods of the code, the associated Jupyter Notebooks provided by the Group should be consulted.

Data Import and Preprocessing

McAuley's aforementioned page was used to access all data for this study. Datasets were downloaded as "reviews_Movies_and_TV_5.json.gz" (approx. 2 GB) and "reviews_Baby_5.json.gz" (approx. 1 GB) for the "Movies and TV" and "Baby" data, respectively. Data was read into PySpark using typical code per Figure 1.

```
df = spark.read.json("reviews_Movies_and_TV_5.json.gz")
```

Figure 1. Importing Data using PySpark

The resulting dataframe has a typical structure per Figure 2, where the columns of interest to the Group are "**reviewText**" and "**overall**", representing the user review text and user rating respectively.

```
[7]: df.show(5)
```

```
+----------+-------+-------+--------------------+-----------+--------------+--------------------+--------------------+--------------+
|      asin|helpful|overall|          reviewText| reviewTime|    reviewerID|        reviewerName|             summary|unixReviewTime|
+----------+-------+-------+--------------------+-----------+--------------+--------------------+--------------------+--------------+
|0005019281| [0, 0]|    4.0|This is a charmin...|02 26, 2008| ADZPIG9QOCDG5|Alice L. Larson "...|good version of a...|    1203984000|
|0005019281| [0, 0]|    3.0|It was good but n...|12 30, 2013|A35947ZP82G7JH|       Amarah Strack|Good but not as m...|    1388361600|
|0005019281| [0, 0]|    3.0|Don't get me wron...|12 30, 2013|A3UORV8A9D5L2E|     Amazon Customer|Winkler's Perform...|    1388361600|
|0005019281| [0, 0]|    5.0|Henry Winkler is ...|02 13, 2008|A1VKW06X1O2X7V|Amazon Customer "...|It's an enjoyable...|    1202860800|
|0005019281| [0, 0]|    4.0|This is one of th...|12 22, 2013|A3R27T4HADWFFJ|                BABE|    Best Scrooge yet|    1387670400|
+----------+-------+-------+--------------------+-----------+--------------+--------------------+--------------------+--------------+
only showing top 5 rows
```

Figure 2. Typical Amazon Review Data

As the "**overall**" scores were discrete values of 1, 2, 3, 4, and 5 it was elected by the Group to filter all items with a score of 3 (or "neutral"). This was chosen in order to reduce ambiguity with how to best classify a review with a "neutral" value, as by its very nature some such reviews may tend to lean positively while others may tend to lean negatively; by contrast, the expectation by the Group is that a review with a 4 or 5 rating value should tend to unambiguously be a "positive" review, while a review with a 1 or 2 rating value should tend to unambiguously be a "negative" review. Further, the Group hoped such removal of reviews scored as a 3 will give a better understanding of how different vocabulary relates to reviews that are clearly positive/negative.

To further prepare the data for NLP, all words were tokenized using **pyspark.ml.feature** methods, with casing dropped and punctuation removed. While stripping punctuation may result in some erroneous cases (such as "she'll" becoming the word "shell" or "can't" becoming the word "cant"), any loss in such fidelity is considered trivial for this study. Stop words (such as articles like "the" or "a") were also removed.

Data and Methods

McAuley's Amazon dataset for the "Baby" category had 160,792 entries, whose mean rating value was 4.21. The "Movies and TV" category had 1,689,188 entries, whose mean rating value was 4.11. The breakdown of the number of reviews per rating value is per the following figure, noting that neutral reviews (i.e. those with a score of 3) have been dropped.

overall	label	count
2.0	0.0	102410
5.0	1.0	906608
1.0	0.0	104219
4.0	1.0	382994

overall	label	count
2.0	0.0	9193
5.0	1.0	93526
1.0	0.0	7819
4.0	1.0	32999

Figure 3. Data Breakdown for "Movies and TV" (left) and "Baby" (right)

Logistic Regression

Logistic Regression was initialized by splitting the data into training (80%) and test (20%) splits for both "Baby" and "Movies and TV" datasets. The Group ran the data through a pipeline that tokenizes the review text, removes stopwords, and forms a term frequency-inverse document frequency (tf-idf) matrix. The Group then ran a Logistic Regression with three-fold cross validation on the training data with the hyperparameters set as follows: the number of features in the model is set at 50000, the regularization parameter is set to 0.10, the elastic net parameter set to 0.10, and the maximum number of iterations to run is set at 10.

Running the Logistic Regression on the "Movies and TV" data gave the following results, rounded to the third decimal point: the average cross-validation accuracy was 0.877. When the cross-validated model was run on the test data the resulting accuracy was 0.879. The F1 score in the test data was found to be 0.838. Running the Logistic Regression on the "Baby" data gave the following results, the average cross-validation accuracy was 0.888. When the cross-validated model was run on the test data the resulting accuracy was 0.891. The F1 score in the test data was found to be 0.847. Neither of these models gave desirable results, but it is interesting that the "Movies and TV" dataset gave better results than the "Baby" dataset. This is probably due to the larger sample size of the "Movies and TV" dataset compared to the "Baby" dataset.

Naive Bayes

Naive Bayes analysis we performed using a similar pipeline to the Logistic Regression approach described above and as elucidated in the Group's supplementary Jupyter Notebook files. Here, the Group used a four-fold cross-validation approach. The hyperparameters were set to the following: the number of features in the model is 40000, and the smoothing parameter was set to 1.0.

Running the Naive Bayes model on the "Movies and TV" data gave the following results, again rounded to three decimal places: the average cross-validation accuracy was found to be 0.833, the accuracy in the test data is 0.832. The F1 score in the test data is 0.852. Running the Naive Bayes model on the "Baby" data gave the following results, the average cross-validation accuracy was 0.865. When the cross-validated model was run on the test data the resulting accuracy was 0.862. The F1 score in the test data was found to be 0.877. Neither of these models give desirable results either, and again note that the model trained on the "Movies and TV" dataset gives better results than the model trained on the "Baby" dataset.

Random Forest

Finally, the Group considered Random Forest for the last model performed on this dataset. The pipeline for this model is similar to those described above, with further detail available in the corresponding Jupyter Notebooks. Here the Group used a four-fold cross-validation approach. Entropy was used as the trait to calculate information gain in the model. The hyperparameters for this model were set according to the following: the number of features used in the model are 50000, the number of trees generated is set at 31, the maximum depth of the trees is set at 29, and the minimum instances per node is set at 1.

Running the Random Forest classifier on the "Movies and TV" data gave the following results: the average cross-validated accuracy was found to be 0.862. The accuracy in the test data was found to be 0.862. The F1 score in the test data was found to be 0.799. Running the Random Forest classifier on the "Baby" data gave the following results: the average cross-validated accuracy was found to be 0.881. The accuracy in the test data was found to be 0.884. The F1 score in the test data was found to be 0.829.

It is thus recognized that the Naive Bayes approach gave slightly better results relative to the other models explored in this study. Like the previous models, the Naive Bayes model trained on the "Movies and TV" dataset produces more comparable results than the "Baby" dataset.

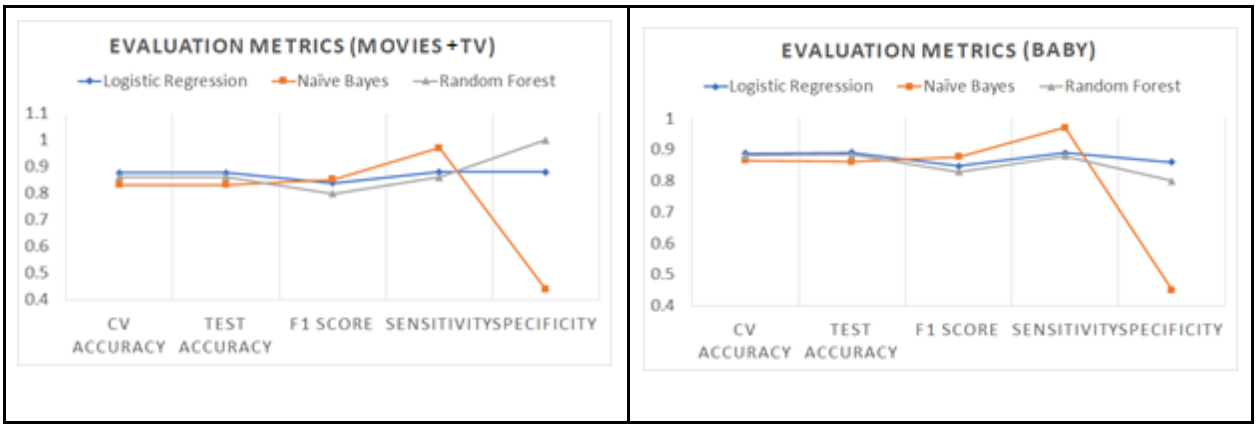

Figure 4. Evaluation Metrics "Movies and TV" (left), "Baby" (right)

Word Clouds

Word clouds were generated to give a sense of the most commonly-occurring words in both the "Movies and TV" and "Baby" datasets, split between the positive and negative reviews respectively. To build the "Movies and TV" word clouds, a random sample of 3-million tokenized words were from the set of positive and negative reviews, respectively. For the "Baby" dataset, all tokenized words were taken for each of the respective positive and negative word clouds.

Figure 5. Typical Positive "Movies and TV" Word Cloud

Figure 6. Typical Negative "Movies and TV" Word Cloud

Figure 7. Typical Positive "Baby" Word Cloud

Figure 8. Typical Negative "Baby" Word Cloud

Results

The data structure of the train/test shows 14% label 0 (negative class) in contrast to 86% label 1 (positive class). This is the evidence that the data is imbalanced as shown in Table 1. The Group adopted F1 score, sensitivity, specificity for model evaluation. CV accuracy and test accuracy are considered as references.

Table 1. Structure of Train/Test Data

	train		test	
TV and movies	label	count	label	count
	0	165559	0	41070
	1	1031561	1	258041
	train		test	
Baby	label	count	label	count
	0	13657	0	3355
	1	101617	1	25458

Table 2. Summary of Amazon Reviews Evaluation Matrix

	TV					Baby				
	CV accuracy	Test accuracy	F1 score	Sensitivity	Specificity	CV accuracy	Test accuracy	F1 score	Sensitivity	Specificity
Logistic Regression	0.877	0.878	0.839	0.88	0.88	0.888	0.891	0.848	0.89	0.86
Naïve Bayes	0.833	0.832	0.852	0.97	0.44	0.865	0.862	0.877	0.97	0.45
Random Forest	0.862	0.863	0.799	0.86	1	0.881	0.884	0.829	0.88	0.8

Table 2 shows a summary of the evaluation matrix. As observed, the Naive Bayes method was found to give higher F1 scores (0.85, 0.88) for predicting sentiment from the user reviews posted on Amazon relative to the corresponding rating score for each of the product categories, “Movies and TV” and “Baby”. The F1 score for the Logistic Regression model gave values of 0.839 and 0.848 respectively, while the F1 score for the Random Forest model gave scores of 0.799 and 0.829 respectively. F1 score for all three algorithms in this study were > 80%.

It was also found in the word clouds that some of the most dominant words associated with positive “Movie and TV” reviews include “film”, “movie”, “good”, and “great”. These words not only describe the products themselves (“film”, “movie”) but include positive sentiments like “good” and “great”. Positive “Baby” reviews feature words like “baby”, “easy”, “great” and “love”. This shows that users put an emphasis on products that are easy to use, with positive sentiment conveyed with words like “great” and “love”, which is similar to what was observed in the “Movie and TV” word cloud.

For the negative word clouds, for “Movie and TV” one can observe words like “film” and “movie”, while also finding fewer dominant words like “good”, “story”, and “plot”. It may be unusual to see the word “good” in the negative word cloud, however without knowing the context it may be such that a negation was used (such as “not good”); further investigation would be needed to determine if building a negative word cloud on unigrams would address this observation. This also suggests that consumers may be focused on media which has quality narratives given the focus on terms like “story” and “plot”.

The negative word cloud for “Baby” saw the word “baby” as dominant, as well as the word “one”. This perhaps indicates that parents of newborns (such as one-year-olds) are more inclined to score products negatively, given the extra care needed for young babies.

Conclusion

Using PySpark was critical in this study for being able to process large amounts of data (on the magnitude greater than 1 GB) efficiently. In this study, the Group found that the three models Logistic Regression, Naïve Bayes and Random Forest achieved F1 score > 80% at predicting sentiment in the Amazon “Movies and TV” and “Baby” combined data sets.

In training sentiment analysis for product reviews or other such social media posts or similar corpora, it would then be recommended that a Naive Bayes approach be considered. Further investigation would be needed to determine how this model would perform for other datasets, not only other Amazon product categories, but also for other review-based platforms such as Yelp or IMDb. Logistic Regression also shows a comparable F1 score with Naive Bayes however, Logistic Regression requires longer computation time.

It would also be worth investigating whether Random Forest can get better results by further increasing the number of trees of optimized tree depths. However, the computation time will increase as the tree size increases. The Group decided to use Naive Bayes based on it is fast computation time and a slightly better F1 score. Amazon reviews in this study (“Movies and TV” and “Baby”) are imbalanced data which are 16% negative and 86% positive. The specificity and sensitivity are also considered for the evaluation matrix. Since the training of 3GB data needs higher computation power, Naive Bayes supports the usage as the preferred algorithm for other similar studies in the future.

The future work is to improve the model performance: the consideration of bi-grams or tri-grams text tokenization are some such options. The token words are essential to the change probability of model prediction. The optimized n-gram as text tokenization is worth evaluating. Other machine learning algorithms such as SVM, KNN are also of interest. Deep learning such as neural networks is a suggestion for sentiment analysis.

References

1. McAuley, Julian. *Amazon Product Data*. http://jmcauley.ucsd.edu/data/amazon/. Accessed 30 July, 2020.

2. Singh, Pramod. *Machine Learning with PySpark: With Natural Language Processing and Recommender Systems.* Berkeley, CA: Apress, 2019.

3. Manovich, Lev. “The Science of Culture? Social Computing, Digital Humanities and Cultural Analytics”. Journal of Cultural Analytics: 10.22148/16.004, 2016.