Exploring Sentiment Analysis in Amazon Product Reviews

DS5559: Big Data Analytics
MSDS, Summer 2020

Group 4
Huilin Chang (hc5hq)
Gavin Wiehl (gtw4vx)
Travis Vitello (tjv9qh)

30 July 2020

Contents

Executive Summary

- Our group sought to perform sentiment analysis on Amazon product reviews
- The question: Could we train a model that accurately predicts the sentiment of an Amazon review based on the user score for that product (positive or negative)?
- Over 2 GB of Amazon product reviews were processed using ML methods, including Logistic Regression, Naïve Bayes, and Random Forest
- It was found that Random Forest performed the best for this data set, while Logistic Regression and Naïve Bayes performed very poorly
- Future studies would intend to build on these results, with refined hyperparameters and alternate datasets

Data Summary

- Data was accessed from Julian McAuley's UCSD data science page
- The data consisted of Amazon product reviews spanning May 1996 through July 2014 and consisted of the "Movie and TV" and "Baby" datasets
- "Movie and TV": 1,689,188 reviews | "Baby": 160,792 reviews
- Each review consisted of 9 columns of data; considered in this study were "overall" (discrete user-given product rating from 1 to 5) and "summary" (user's review text)

```
[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

Typical Amazon Review Data

Variable Transformations and Preprocessing

- Reviews were grouped into “positive” (review scores of 4 or 5) and “negative” (review scores of 1 or 2); review scores of 3 were dropped to eliminate ambiguity
- Text was cleaned by dropping punctuation and case to allow tokenization via pyspark.ml.feature methods
- The breakdown of the reviews by individual dataset are seen the below figure

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

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

Model Performance

Logistic Regression

Hyperparameters: num of features set at 50,000, regularization parameter set to 0.10, elastic net parameter set to 0.10, max num of iterations set at 10.

On the “Movies and TV” dataset:

- Average 3-fold cross validation accuracy was 0.88.
- Test holdout accuracy was 0.87
- F1 score in the test holdout was 0.838.

On the “Baby” dataset:

- Average 3-fold cross-validation accuracy was 0.89.
- Test holdout the accuracy was 0.89.
- F1 score in the test data was found to be 0.847.

Logistic Regression confusion matrix and classification report

Movies and TV

```
average cross-validation accuracy = 0.8772973190859461
Accuracy in the test data = 0.8789379193677264
F1 score in the test data = 0.8385959162917399
              precision    recall  f1-score   support

       neg 0       0.14      0.88      0.24      6427
       pos 1       1.00      0.88      0.93    292684

    accuracy                           0.88    299111
   macro avg       0.57      0.88      0.59    299111
weighted avg       0.98      0.88      0.92    299111

[[  5643    784]
 [ 35427 257257]]
```

	Negative	Positive
Negative	5643	784
Positive	35427	257257

Baby

```
average cross-validation accuracy = 0.888054075144795
Accuracy in the test data = 0.8906396418283413
F1 score in the test data = 0.8475512158454273
              precision    recall  f1-score   support

       neg 0       0.07      0.86      0.13       282
       pos 1       1.00      0.89      0.94     28531

    accuracy                           0.89     28813
   macro avg       0.54      0.88      0.54     28813
weighted avg       0.99      0.89      0.93     28813

[[  243    39]
 [ 3112 25419]]
```

	Negative	Positive
Negative	243	39
Positive	3112	25419

Model Performance

Naïve Bayes

Hyperparameters: num of features set at 40,000, the smoothing parameter set to 1.0.

On the “Movies and TV” dataset:

- Average 3-fold cross-validation accuracy was 0.833.
- Test holdout accuracy was 0.832.
- F1 score in the test holdout was 0.852.

On the “Baby” dataset:

- Average 3-fold cross-validation accuracy was 0.865.
- Test holdout the accuracy was 0.862.
- F1 score in the test data was found to be 0.877.

Naive Bayes confusion matrix and classification report

Movies and TV

```
average cross-validation accuracy = 0.8330808039762652
Accuracy in the test data = 0.8321793581646947
F1 score in the test data = 0.8517766697496387
              precision    recall  f1-score   support

       neg 0       0.84      0.44      0.58     78133
       pos 1       0.83      0.97      0.90    220978

    accuracy                           0.83    299111
   macro avg       0.84      0.71      0.74    299111
weighted avg       0.83      0.83      0.81    299111

[[ 34503  43630]
 [  6567 214411]]
```

	Negative	Positive
Negative	34503	43630
Positive	6567	214411

Baby

```
average cross-validation accuracy = 0.8650649351695062
Accuracy in the test data = 0.862145559296151
F1 score in the test data = 0.8771130624334371
              precision    recall  f1-score   support

       neg 0       0.77      0.45      0.57      5809
       pos 1       0.87      0.97      0.92     23004

    accuracy                           0.86     28813
   macro avg       0.82      0.71      0.74     28813
weighted avg       0.85      0.86      0.85     28813

[[ 2596  3213]
 [  759 22245]]
```

	Negative	Positive
Negative	2596	3213
Positive	759	22245

Model Performance

Random Forest

Hyperparameters: num of features set at 50,000, num of trees set at 31, max depth of the trees set at 29, min instances per node set at 1.

On the “Movies and TV” dataset:

- Average 3-fold cross-validation accuracy was 0.862
- Test holdout accuracy was 0.863.
- F1 score in the test holdout was 0.799.

On the “Baby” dataset:

- Average 3-fold cross-validation accuracy was 0.880.
- Test holdout the accuracy was 0.883
- F1 score in the test data was found to be 0.829.

Random Forest confusion matrix and classification report

Movies and TV

```
average cross-validation accuracy = 0.861708939972216
Accuracy in the test data = 0.8627064868894825
F1 score in the test data = 0.7991328831993197
              precision    recall  f1-score   support

       neg 0       0.00      1.00      0.00         4
       pos 1       1.00      0.86      0.93    299107

    accuracy                           0.86    299111
   macro avg       0.50      0.93      0.46    299111
weighted avg       1.00      0.86      0.93    299111

[[     4      0]
 [ 41066 258041]]
```

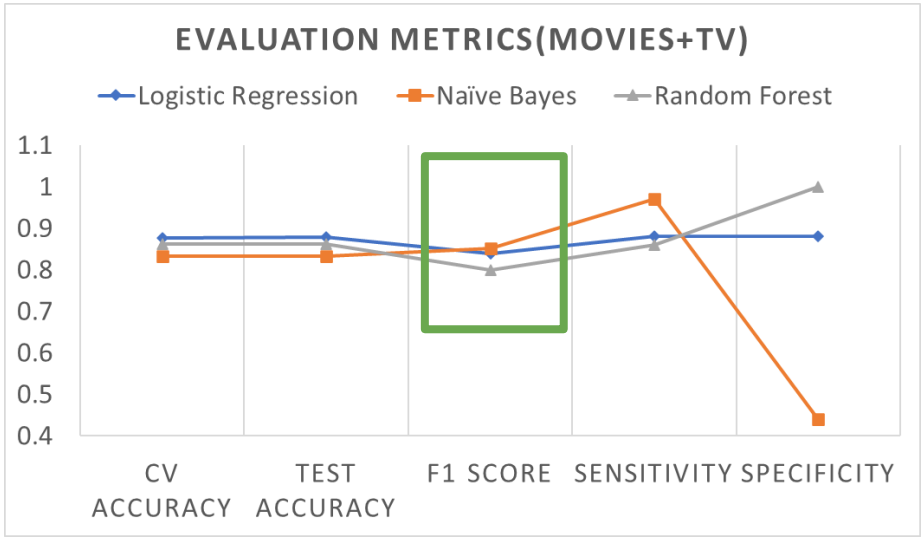
	Negative	Positive
Negative	4	0
Positive	41066	258041

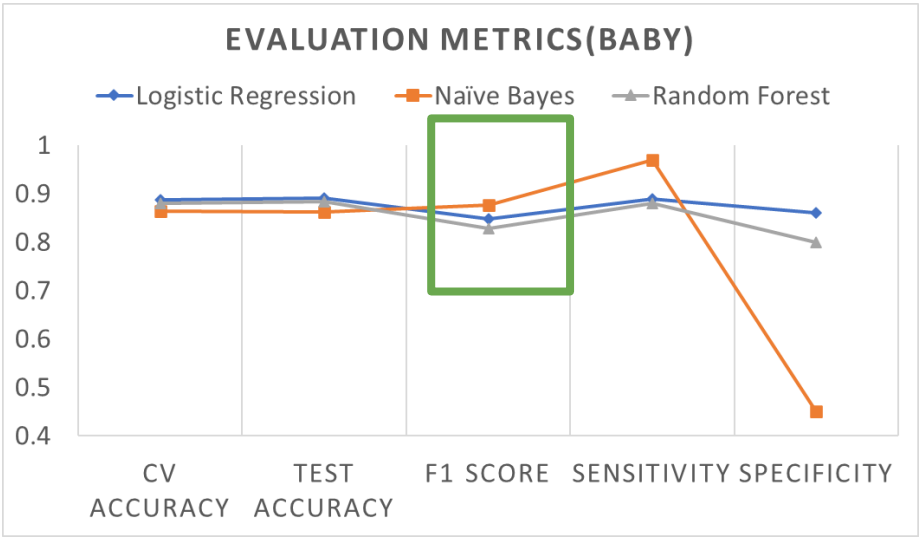
Baby

```
average cross-validation accuracy = 0.8809952833735155
Accuracy in the test data = 0.8836636240585847
F1 score in the test data = 0.8292594492507848
              precision    recall  f1-score   support

       neg 0       0.00      0.80      0.00         5
       pos 1       1.00      0.88      0.94     28808

    accuracy                           0.88     28813
   macro avg       0.50      0.84      0.47     28813
weighted avg       1.00      0.88      0.94     28813

[[    4     1]
 [ 3351 25457]]
```

	Negative	Positive
Negative	4	1
Positive	3351	25457

Evaluation matrix summary charts of three models





It was decided to use Naive Bayes due to less computation time and its slightly better F1 score

Word Clouds

- Word clouds of the individual datasets were generated in order to observe word prevalence in the positive and negative reviews by category studied
- For “Movies and TV”, a random sample of 3 million words from positive and negative reviews (respectively) were taken to improve computation time
- For “Baby”, the complete set of words was taken
- Words were taken as unigrams
- All stop words were eliminated by applying the pyspark.ml.feature method StopWordsRemover

Word Clouds (“Movies and TV”)

Typical Positive "Movies and TV" Word Cloud

Typical Negative "Movies and TV" Word Cloud

Word Clouds (“Baby”)

Typical Positive “Baby" Word Cloud

Typical Negative “Baby" Word Cloud

Conclusions and Future Research

- Both Amazon reviews (Movies and TV, Baby) are imbalanced data; therefore F1 score for the model selection was adopted as the evaluation criterion
- It was decided to use Naive Bayes for model selection due to less computational time and slightly better F1 score of methods explored (85%, 88%)
- Random Forest shows an F1 score (80%, 83%), the computation time needed is very challenging for this big data set (> 4hrs)
- The possible reasons for Random Forest's performance were optimized tree depth/number of trees that are required to achieve a better model performance
- It's noted that the Group expected Random Forest to perform the best, which may be possible with refined parameter choices

Conclusions and Future Research

- Future research would consider alternate ML methods such as SVM or clustering algorithms like K-means; also, refining parameters used in previous methods
- Other research may also consider alternative datasets, including other Amazon product reviews but also reviews from other platforms like Yelp or IMDb
- Finally, future research may also consider studying the correlation of user sentiment across several products (if a Person likes / dislikes X, do they like / dislike Y?)
- This may also be extended to see how product sentiment changes over time


Questions?