CAPSTONE PROJECT

CUSTOMER REVIEW – SENTIMENT GENERATION

BY
MANASA S

TABLE OF CONTENTS

PROBLEM STATEMENT

An e-commerce company has collected the feedback of different items from customers and also their helpfulness indices. They want to analyze the customer reviews for various products and judge if the reviews are positive or negative.

The aim is to create a report that classifies the products based on the customer reviews.

PROJECT OBJECTIVE

1. It is required to classify products based on the customer sentiments recorded.
2. Find various trends and patterns in the data; create useful insights that best describe the product quality.
3. Analyze the products based on reviews.

DATA DESCRIPTION

The dataset 'REVIEW.csv' consists of 568454 rows and 10 columns.

S.No	FEATURE NAME	DESCRIPTION
1.	Id	Record ID
2.	Product Id	Product ID
3.	User Id	User ID who posted the review
4.	Profile Name	Profile name of the User
5.	Helpfulness Numerator	Numerator of the helpfulness of the review
6.	Helpfulness Denominator	Denominator of the helpfulness of the review
7.	Score	Product Rating
8.	Time	Review time in timestamp
9.	Summary	Summary of the review
10.	Text	Actual text of the review.

DATA PREPROCESSING STEPS AND INSPIRATION

1. **EDA:**

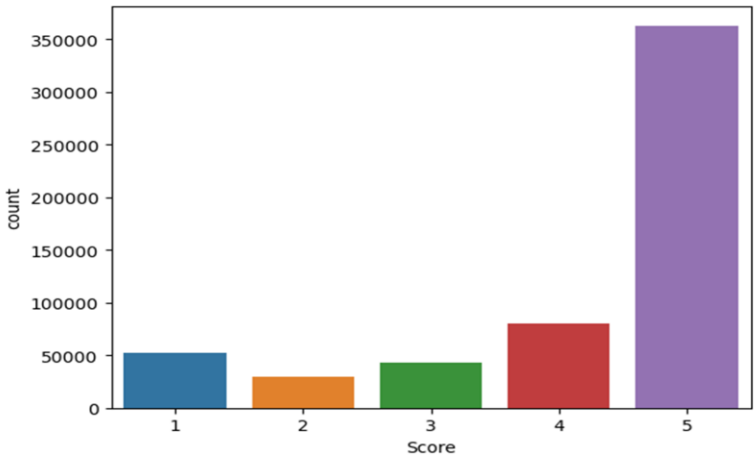
(a) The unique records in each column are noted. There are only 74258 unique products and 256047 unique users.

Id	568411
ProductId	74258
UserId	256047
ProfileName	218413
HelpfulnessNumerator	231
HelpfulnessDenominator	234
Score	5
Time	3168
Summary	295736
Text	393565

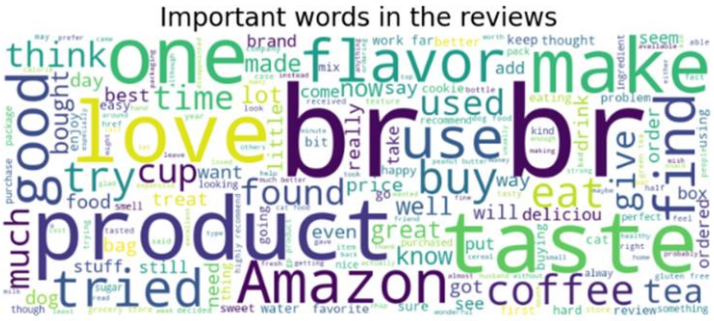
(b) By grouping the products based on average rating, it is noted that most of the products have an average rating of 5.

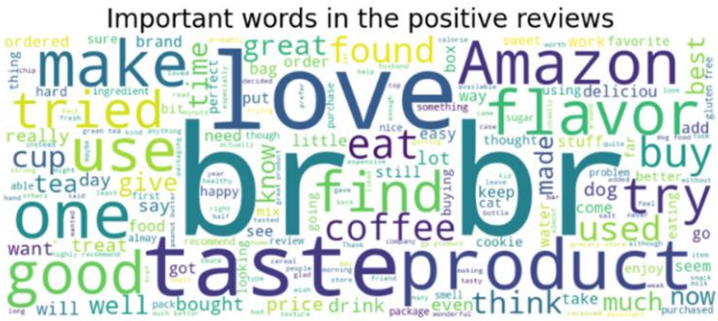
Average rating	
1.000000	4118
1.062500	16
1.142857	7
1.200000	15
1.250000	36
...	
4.964286	84
4.968750	64
4.972973	37
4.973451	113
5.000000	52787

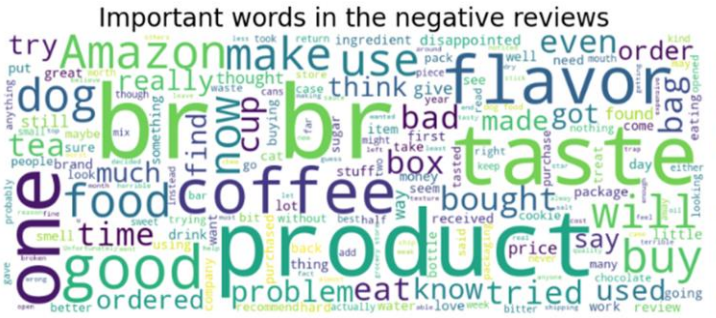
(c) Most of the items have a score of 5.

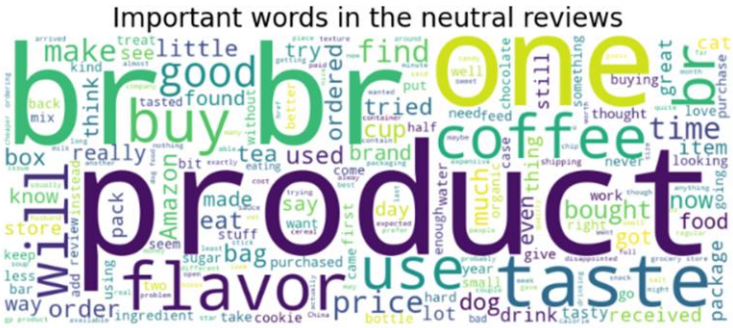


(d) The most commonly word used in general text, positive texts, negative and neutral texts were observed using word cloud.









We note that most of the words used are not capturing the sentiment and hence the frequency of words that express sentiment is captured.

(e) Most of the positive and negative reviews can be tagged as not helpful as the helpfulness numerator is 0

	Positive reviews:		Negative reviews:
0	269229	0	28142
1	101306	1	12671
2	46309	2	6261
3	24786	3	3570
4	14261	4	2258

2. There were 16 null values in profile name and 27 null values in summary column. Since the records are very small when compared to the total number of records, we drop them. Now the data reduced to 568411 records.

```
Id                          0
ProductId                   0
UserId                      0
ProfileName                16
HelpfulnessNumerator        0
HelpfulnessDenominator      0
Score                       0
Time                        0
Summary                    27
Text                        0
```

```
1 data.shape
```

```
(568411, 10)
```

3. The data is checked for any duplicated values. Here there are no duplicates.

```
1 data.duplicated().sum()
```

```
0
```

4. Average rating columns is created which gives the average of the scores given by all the customers for a given product.

Score	Time	Summary	Text	Average rating
5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...	5.00
1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...	1.00
4	1219017600	"Delight" says it all	This is a confection that has been around a fe...	4.00
2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...	2.00

5. The given text reviews were cleaned using spacy.

Text	Average rating	Sentiment compound	Sentiment compound summary	Sentiment	polarity	subjectivity	Sentiment_textblob	target	text_clean
i have bought several of the vitality canned d...	5.00	0.9441	0.4404	Positive	0.450000	0.433333	Positive	Positive	I have buy several of the vitality can dog foo...
product arrived labeled as jumbo salted peanut...	1.00	-0.5664	0.0000	Negative	-0.033333	0.762963	Neutral	Negative	product arrive label as jumbo salt peanut ... ...
this is a confection that has been around a fe...	4.00	0.8265	0.0000	Positive	0.133571	0.448571	Positive	Positive	this be a confection that have be around a few...
if you are looking for the secret ingredient i...	2.00	0.0000	0.0000	Neutral	0.166667	0.533333	Positive	Negative	if you be look for the secret ingredient in ro...

CHOOSING THE ALGORITHM FOR THE PROJECT

The given data was analyzed for sentiment generation as well as for sentiment classification.

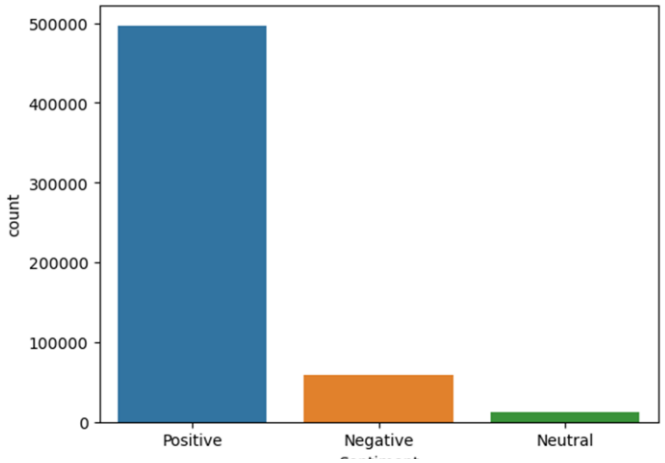
1. VADER SENTIMENT GENERATOR:

Using sentiment intensity analyzer of Vader, the compound was captured and based on the compound generated, a function classified it into positive or negative or neutral sentiments.

	Score	Time	Summary	Text	Average rating	Sentiment compound	Sentiment compound	Sentiment compound summary	Sentiment
5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...	5.00	0.9441	0.9441	0.4404	Positive
1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...	1.00	-0.5664	-0.5664	0.0000	Negative
4	1219017600	"Delight" says it all	This is a confection that has been around a fe...	4.00	0.8265	0.8265	0.0000	Positive
2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...	2.00	0.0000	0.0000	0.0000	Neutral
						0.9468	0.6249	Positive

It is noted that there are more positive sentiments than others.

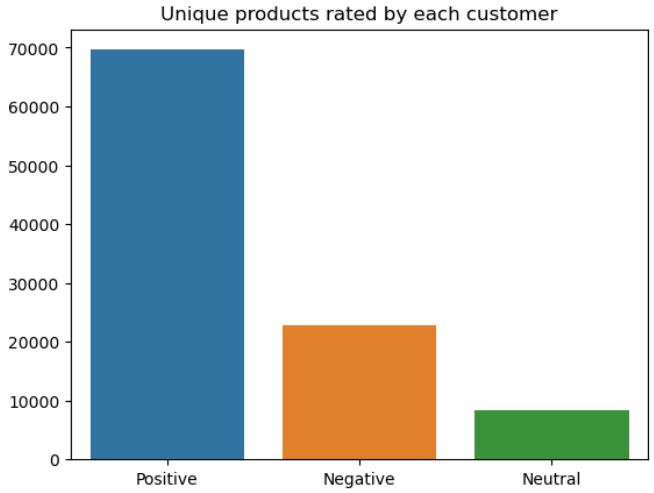
```
Positive    496938
Negative     59303
Neutral      12170
Name: Sentiment, dtype: int64
```



Totally 69646 unique products have been rated positive

22891 unique products have been rated negative

8268 unique products have been rated neutrally by users.



2. TEXT BLOB:

Using Text Blob, polarity scores, the data was categorized into Positive, negative and neutral sentiments by the same function defined above.

Sentiment	polarity	subjectivity	Sentiment_textblob
Positive	0.450000	0.433333	Positive
Negative	-0.033333	0.762963	Neutral
Positive	0.133571	0.448571	Positive
Neutral	0.166667	0.533333	Positive
Positive	0.483333	0.637500	Positive

It is noted that there are more Positive scores than negative or neutral ones.

```
Positive    473146
Neutral      55584
Negative     39681
Name: Sentiment_textblob
```

3. The data was cleaned using spacy's lemmatizer and the data was tr ansformed to 568411x108068 sparse matrix. Extra features can be added to the matrix of independent variables. The dependent vari able column has been formed by mapping the sentiments to a nu mber. This data can now be passed through any of the ML classific ation algorithms.

ASSUMPTIONS

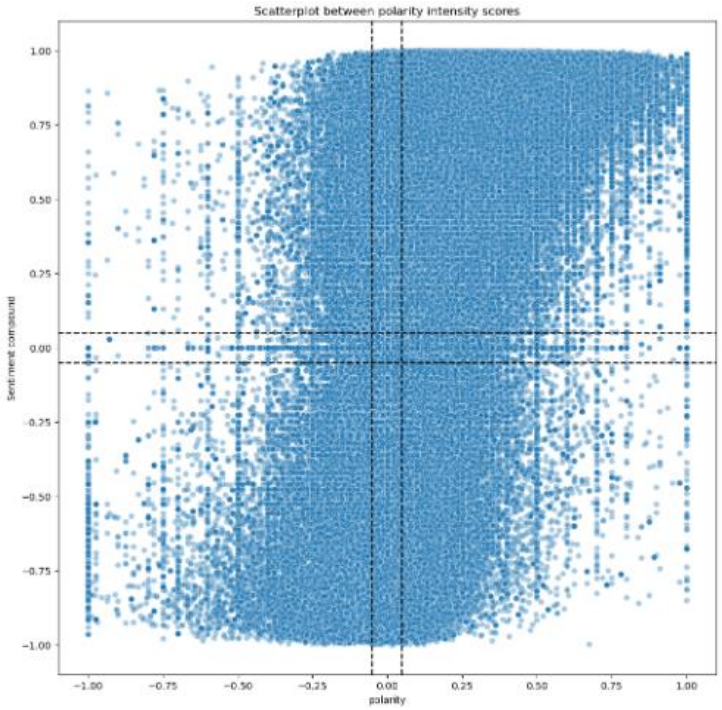
1. Reviews provided by the users are independent and identically distributed.
2. There is no intended sarcasm or idioms and emojis which are difficult to interpret.

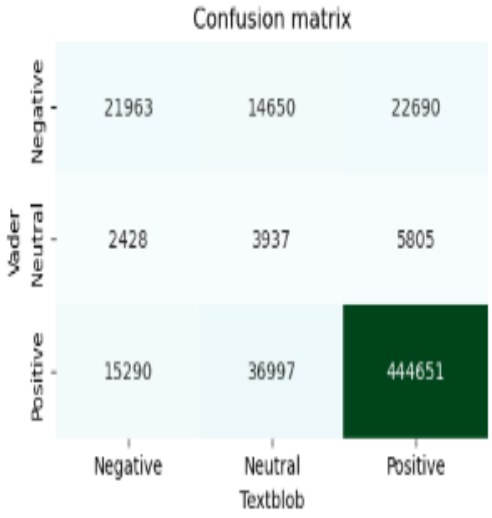
MODEL EVALUATION AND TECHNIQUE

The sentiment classification model can be evaluated using precision, recall, f1 score and accuracy.

The sentiment generator has been compared with the performance of the other.

1. Vader and text blob's performances were compared and it is noted that though many points which were classified by Text blob goes hand in hand with Vader, good amount of points which Vader has classified positive has been classified otherwise by Text blob and vice versa.





Here target column which is sentiment based on the average rating of a product can also be used as a base to compare both the scores.

2. The sparse matrix can be fit into any ML algorithms and analyzed for its ability to capture the sentiment correctly. Metrics used are accuracy score, F1 score, recall and precision.

INFERENCES FROM THE PROJECT

1. The given text reviews were cleaned and sentiment analyzer applied to the data. It is observed that both Vader and Text Blob do not give the same result completely, though it is same for most of the points.
2. The data is heavily biased towards positive reviews in both of the above cases and hence the results obtained are imbalanced.

```
1 data['Sentiment'].value_counts()
```

```
Positive    496938
Negative     59303
Neutral      12170
Name: Sentiment, dtype: int64
```

```
1 data['Sentiment_textblob'].value_counts()
```

```
Positive    473146
Neutral      55584
Negative     39681
Name: Sentiment_textblob, dtype: int64
```

3. With the imbalanced dataset, if we fit a ML model then the scores are bad.

```
TRAINING SCORE:

Accuracy score:  0.9275632905825021

Classification report:
              precision    recall  f1-score   support

          -1       0.79      0.59      0.67     47411
           0       0.92      0.24      0.39      9734
           1       0.94      0.98      0.96    397583

    accuracy                           0.93    454728
   macro avg       0.89      0.61      0.67    454728
weighted avg       0.92      0.93      0.92    454728
```

```
TESTING SCORE:

Accuracy score:  0.921052400095001

Classification report:
              precision    recall  f1-score   support

          -1       0.76      0.56      0.65     11892
           0       0.84      0.18      0.30      2436
           1       0.93      0.98      0.96     99355

    accuracy                           0.92    113683
   macro avg       0.84      0.57      0.63    113683
weighted avg       0.91      0.92      0.91    113683
```

We see that in the testing scores:

(i) Negative class has good precision but low recall indicating that the model has poor ability to detect a negative class point but at the same time when it does its trustable to an extent.

(ii) Neutral class has good precision but very less recall and this su ggests that the model is very weak in identifying a neutral class bu t at the same time when it does, its trustable to an extent.

(iii) Positive class has a good precision and recall suggesting that the model has fit well over the data.

This may be due to imbalanced dataset.
Hence under-sampling was performed and the models were trained using the new data points.

(a) Logistic Regression:

--------- LogisticRegression() ---------

TRAINING SCORE:

Accuracy score: 0.854354971240756

Classification report:

	precision	recall	f1-score	support
0	0.84	0.83	0.83	9787
1	0.83	0.82	0.83	9690
2	0.89	0.91	0.90	9731
accuracy			0.85	29208
macro avg	0.85	0.85	0.85	29208
weighted avg	0.85	0.85	0.85	29208

TESTING SCORE:

Accuracy score: 0.7937551355792933

Classification report:

	precision	recall	f1-score	support
0	0.77	0.75	0.76	2383
1	0.76	0.75	0.76	2480
2	0.85	0.88	0.87	2439
accuracy			0.79	7302
macro avg	0.79	0.79	0.79	7302
weighted avg	0.79	0.79	0.79	7302

Though accuracy has decreased, it's still better as precision f1 score and recall are comparable.

(b) Random Forest classifier:

TRAINING SCORE:

Accuracy score: 0.9887017255546425

Classification report:

	precision	recall	f1-score	support
0	1.00	0.98	0.99	9787
1	0.97	1.00	0.98	9690
2	1.00	0.99	0.99	9731
accuracy			0.99	29208
macro avg	0.99	0.99	0.99	29208
weighted avg	0.99	0.99	0.99	29208

TESTING SCORE:

Accuracy score: 0.8155299917830732

Classification report:

	precision	recall	f1-score	support
0	0.81	0.74	0.78	2383
1	0.77	0.84	0.80	2480
2	0.87	0.86	0.87	2439
accuracy			0.82	7302
macro avg	0.82	0.81	0.81	7302
weighted avg	0.82	0.82	0.82	7302

With this model scores are better than the Logistic regression models. But the model has overfit the data since training accuracy is very high than the testing one.

(c) XGBoost Classifier:

TRAINING SCORE:

Accuracy score: 0.9067036428375788

Classification report:

	precision	recall	f1-score	support
0	0.90	0.87	0.89	9787
1	0.87	0.91	0.89	9690
2	0.95	0.93	0.94	9731
accuracy			0.91	29208
macro avg	0.91	0.91	0.91	29208
weighted avg	0.91	0.91	0.91	29208

TESTING SCORE:

Accuracy score: 0.8160777869076965

Classification report:

	precision	recall	f1-score	support
0	0.77	0.77	0.77	2383
1	0.80	0.79	0.80	2480
2	0.88	0.88	0.88	2439
accuracy			0.82	7302
macro avg	0.82	0.82	0.82	7302
weighted avg	0.82	0.82	0.82	7302

Here the data is comparable in terms of the scores obtained. Though the obtained model is not the best, we can hypertune the parameters to get an efficient model

FUTURE POSSIBILITIES

The models can be used with fresh incoming data points in future and the best algorithm can be chosen based on the Recall, Precision and F1 scores. Hyper parameter tuning can be performed on the model to get a better model. Also additional columns which might impact the reviews of the customers can be added to the existing sparse matrix to train the model better. Sample weights can be used as well rather than doing undersampling.

CONCLUSION

In this project the customer segmentation was analyzed with a given data using Vader analyzer and text blob. The two results were compared and good number of them was misclassified. Also sparse matrix is obtained from the data to build sentiment generator for incoming data as well, after undersampling or by assigning sample weigths.

REFERENCES

1. Handling imbalanced datasets in machine learning | by Baptiste Rocca | Towards Data Science

2. Python Word Clouds Tutorial: How to Create a Word Cloud | DataCamp

3. VADER Vs. TextBlob — Which One Is Better For Social Media Sentiment Analysis? | by Zoumana Keita | Geek Culture | Medium