Sentiment Analysis - Cell_Phones_and_Accessories

Group 3: Aneesh Nair$^{\#1}$, Mit Patel$^{\#2}$, Punya Sridharan$^{\#3}$, Darshana Khadke$^{\#4}$

$^{\#}$IMSE5570-Introduction to Big Data (Fall-2020)
Professor – Prof. Mohamed Abouelenien

***Abstract*— We examine sentiment analysis on Amazon reviews for cell phone categories of product. The aim of this project is to classify each review into a positive or negative class of sentiment. We have applied the Logistic regression algorithm to predict the class of review. The data model was validated using standard metrics, such as accuracy, ROC and F1-score.**

***Keywords*— Sentiment Analysis, Amazon Reviews, Logistic Regression, Data Cleaning, Cell Phone**

I. Introduction

Sentiment analysis is contextual mining of text which identifies and extracts subjective information in the source material and helps a business to understand the social sentiment of their brand, product, or service while monitoring online conversations. It is one of the most common text analysis tools which predicts or classifies if the underlying tone of an incoming text is positive, neutral, or negative sentiment.

In this project, we have attempted to predict the sentiments of amazon products review for the Cell Phones and Accessories categories. The data set is of JSON format that consists of reviews from 2002 to 2018. For this project, we have filtered out non validated reviews and used only validated reviews for the data model

Logistic regression:

Logistic regression is a predictive analysis algorithm that is used for classification problems. To map predicted values to probabilities, we use the Sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities. The hypothesis of logistic regression tends to limit the cost function between 0 and 1. Therefore linear functions fail to represent it as it can have a value greater than 1 or less than 0 which is not possible as per the hypothesis of logistic regression

II. Related work

A. (Selvaraj, A Beginner's Guide to Sentiment Analysis with Python 2020) [1]

The article demonstrates an example of sentiment analysis in python. First, the article shows how to read a data frame and visualize the results using graphs and word clouds. Using this article as a reference, we have implemented word clouds and graphs for our visualizations as well. The article also demonstrates the classification of tweets as positive or negative. This was the inspiration for using polarity classification in our project. Finally, based on the article we will also use logistic regression as our algorithm.

B. (Monsters, Text Preprocessing in Python: Steps, Tools, and Examples 2018) [2]

This article discusses the steps to preprocess text. Some of the steps it discusses are the following:

- converting everything to lowercase
- removing numbers
- removing punctuation
- removing white space
- removing stopwords

The article goes into detail about each one of these steps and how to complete them. We will be using this as a reference to complete our preprocessing.

III. Data Description

Data set - Cell_Phones_and_Accessories_5.json
Link- https://nijianmo.github.io/amazon/index.html

1. overall: Rating given by the customer on the scale of 1 to 5. 1 being highly dissatisfied and 5 being highly satisfied
2. verified: If review is verified or not with values True or False
3. reviewTime: Date of the review.
4. reviewerID: Unique ID of the customer
5. asin: Unique Id of the product
6. style: Style of the product
7. reviewerName: Individual person to give a review.
8. reviewText: Detail description of the review.
9. summary: Summary of the review.
10. unixReviewTime: System time of the review.
11. vote: How many other uses find this review helpful or not.
12. image: Images of the product.

IV. Methodology

A. Data Cleaning

Data cleaning aims to reduce our word corpus and have words that will help us predict the sentiment of the text. The most important text field we need for our analysis if review text, hence we will be focusing on cleaning and preparing the field for vectorization

Techniques use - basic python packages such as pandas, NumPy and Data Frame

Total reviews before data cleaning - 1,128,437

1. Checked the data types of each column
2. Converted the reviewDate column to the Date time field
3. Checked for values count of verified columns to use only verified reviews
4. Type conversion has been performed for date and overall column.
5. Dropped unnecessary columns.
6. Remove all the special characters and numbers : Eg: !@#$%^&*()
7. Removing stopping words from reviewText like is an a the which are frequent in a corpus
8. Converting all the words in the corpus to lower case
9. Checking the maximum and minimum length of each reviewText
10. Checking the counts of the rating values in the data set
11. Creating another Data frame with rows that are only validated

Total reviews after data cleaning - 986,589

B. Data Pre-processing & visualization

To predict the sentiment of reviewText which is of the categorical label The Machine Learning algorithm will be of classification model For the model to predict the class, we need to convert the reviewText to numerical values which will be fed as input to the model for classification. This was achieved using NLP techniques such as Tokenization and stemming.

- Tokenization is used to read the text that will be mined and removes all tabs and punctuations between words and replaces them with white space,
- Steaming is used to return all the words to their basic forms where it will remove the plural ‘s’ from the nouns and the ‘ing’ from the verbs.

Before applying the technique we need to understand the overall corpus.

1. Plot the rating of the product into an histogram

It is safe to assume that most of the review text is in the positive tone due to the distribution of the rating.

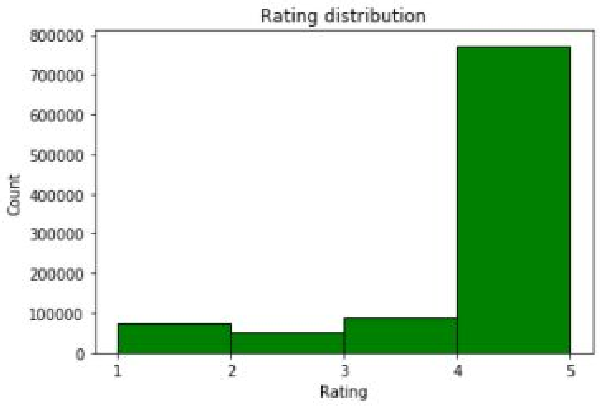


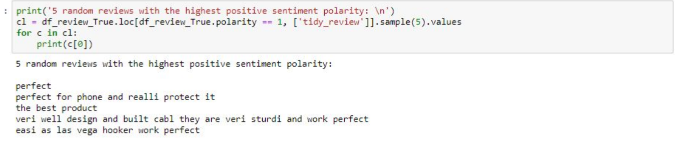
Figure-1

2. Create a polarity column which reviewText

It creates a float value that lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement. Since we do not have a predefined label field we will be assuming that polarity of negative value if of negative sentiment and that of a positive if of positive sentiment.

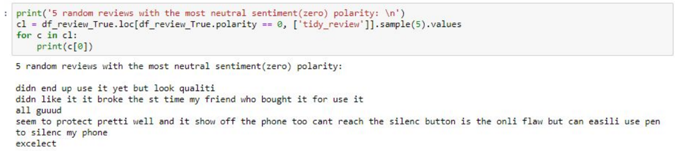
Testing Polarity:

Random 5 texts with positive, neutral, and negative polarity



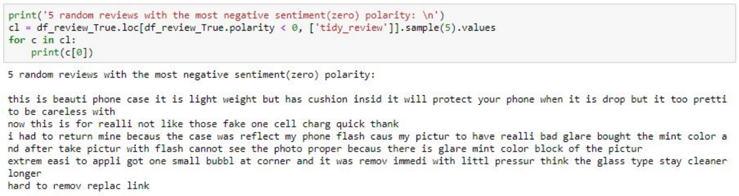
```
print('5 random reviews with the highest positive sentiment polarity: \n')
cl = df_review_True.loc[df_review_True.polarity == 1, ['tidy_review']].sample(5).values
for c in cl:
    print(c[0])
```

```
5 random reviews with the highest positive sentiment polarity:

perfect
perfect for phone and realli protect it
the best product
veri well design and built cabl they are veri sturdi and work perfect
easi as las vega hooker work perfect
```



```
print('5 random reviews with the most neutral sentiment(zero) polarity: \n')
cl = df_review_True.loc[df_review_True.polarity == 0, ['tidy_review']].sample(5).values
for c in cl:
    print(c[0])
```

```
5 random reviews with the most neutral sentiment(zero) polarity:

didn end up use it yet but look qualiti
didn like it it broke the st time my friend who bought it for use it
all guuud
seem to protect pretti well and it show off the phone too cant reach the silenc button is the onli flaw but can easili use pen
to silenc my phone
excelect
```



```
print('5 random reviews with the most negative sentiment(zero) polarity: \n')
cl = df_review_True.loc[df_review_True.polarity < 0, ['tidy_review']].sample(5).values
for c in cl:
    print(c[0])
```

```
5 random reviews with the most negative sentiment(zero) polarity:

this is beauti phone case it is light weight but has cushion insid it will protect your phone when it is drop but it too pretti
to be careless with
now this is for realli not like those fake one cell charg quick thank
i had to return mine becaus the case was reflect my phone flash caus my pictur to have realli bad glare bought the mint color a
nd after take pictur with flash cannot see the photo proper becaus there is glare mint color block of the pictur
extrem easi to appli got one small bubbl at corner and it was remov immedi with littl pressur think the glass type stay cleaner
longer
hard to remov replac link
```

3. Plot the polarity column in histogram to understand the distribution of our reviewText

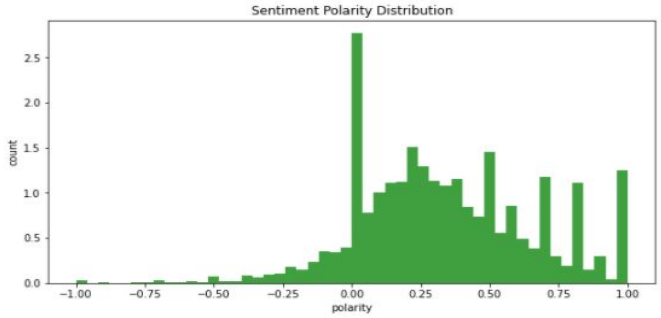


Figure -2

4. Rating Counts -

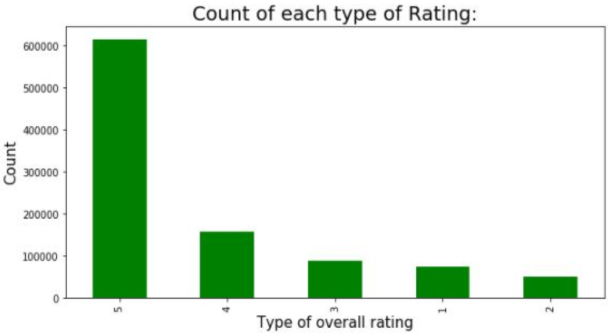


Figure - 3

5. Top 50 words in the corpus -

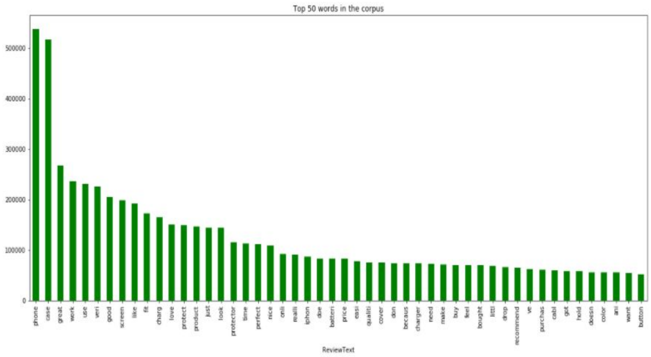


Figure - 4

6. Word Cloud- Creating word cloud to visualize the words in the corpus -

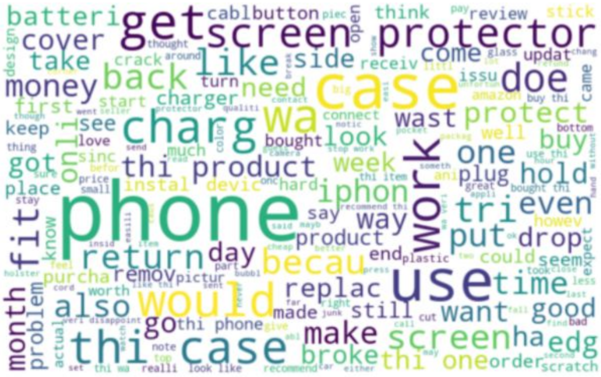


Figure - 5

7. Tokenization and stemming

Used SnowballStemmer from the nltk package to stem the words. The reviewText was tokenized and stemming was applied on the tokenized word. For exam words like excellent would be stemmed to excel. This way we are reducing the words to their root form while keeping the actual meaning and tone of the review text intact.

8. Creating labels

We have labeled the reviewText as 0 and 1 based on the polarity value as the predefined label was not present in the data set. If the polarity value is >0 then its positive tone class 1, else its negative tone, class 0. Based on the polarity, we created positive and negative word clouds. What we noticed was there was not a noticeable difference in the two-word clouds.

C. TF-IDF and Count Vectorization

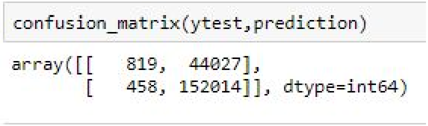
In order, to implement our data into the linear regression algorithm, we need to apply vectorization. Vectorization allows computers to "understand" words by converting them into numbers. Popular techniques to accomplish this are TF-IDF and Count Vectorization. TF-IDF considers the importance of a word in documents. Count Vectorization counts the number of times a word appears in a document. Count Vectorization is often considered the inferior vectorization method due to its ability to be biased towards most frequent words. It prevents rare words that hold more "weight" from representing the document. TF-IDF accounts for these shortcomings. Due to this reason, we used TF-IDF for vectorization.

V. Experimentation

We have used Logistic Regression to train and test our model. The dataset was split into Train and Test using the train_test_split(). 80% - Train set and 20% as a Test set. The model was trained using the Train set (XTrain) and the labels were predicted for XTest. The build data model was evaluated using metrics like Precision, F1 Score, and ROC curve.

1. Confusion Matrix

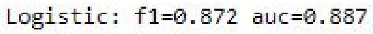
Looking at the confusion matrix, we can see the model has high False Positive and True Positive which implies that the model is not able to differentiate between positive and negative sentiment that well. The reason for this is, most of the review text is more on the positive and neutral sentiment than on the negative sentiment.



```
confusion_matrix(ytest,prediction)

array([[   819,  44027],
       [   458, 152014]], dtype=int64)
```

2. Precision Vs Recall

We got an Accuracy of 0.887 and an F1 score of 0.872 which identifies our True positive cases are very high. It is because of the positive sentiment in our data. F1-score is the harmonic mean of Precision and Recall and gives a better measure of the incorrectly classified cases than the Accuracy Metric.



```
Logistic: f1=0.872 auc=0.887
```

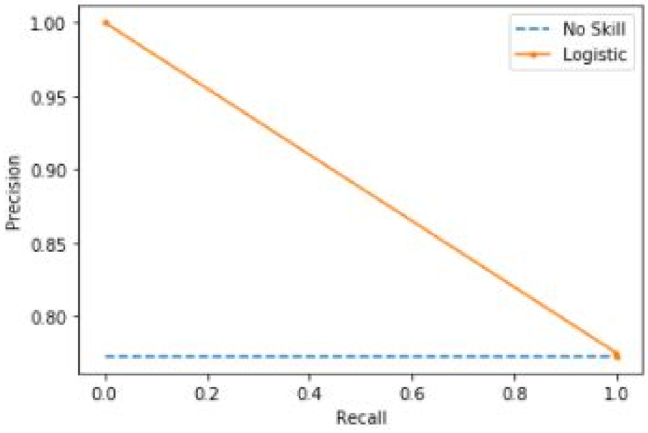
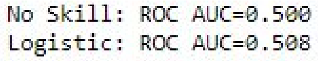


Figure - 6

3. ROC AUC Curve

Our ROC (Receiver or operating characteristic curve) illustrates that model struggling to identify the correct negative sentiment and unable to differentiate the class well.



```
No Skill: ROC AUC=0.500
Logistic: ROC AUC=0.508
```

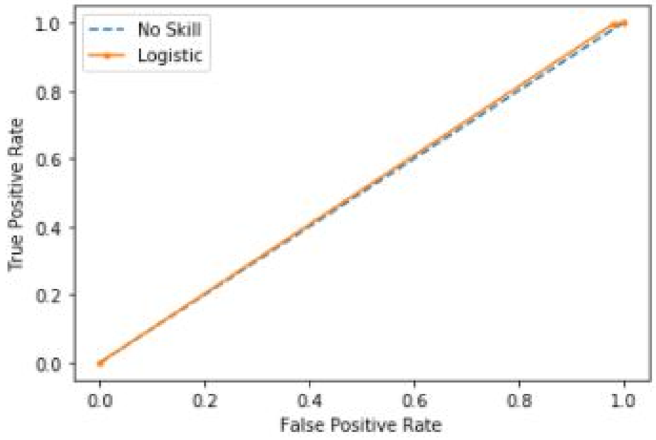


Figure - 7

VI. Contribution

There contribution of each member is represented by the table below:

- Data Cleaning - Darshana Khadke
- Visualization- Aneesh Nair
- Building the model- Mit Patel
- Metrics & predicting the sentiment- Punya Sridharan
- Report preparation -All

VII. Conclusion

Given the trend of the data set, which is more on the positive sentiment and very little % of data on negative sentiment, we were able to build a model with Accuracy 88% ,F1 Score 87% and ROC curve (50%) which implies the model did not perform well. Thus we conclude that reviews were more on the positive and neutral sentiment and less on the negative sentiment.

VIII. references

[1] N. Selvaraj, “A Beginner's Guide to Sentiment Analysis with Python,” Medium, 12-Sep-2020. [Online]. Available: https://towardsdatascience.com/a-beginners-guide-to-sentiment-analysis-in-python-95e354ea84f6. [Accessed: 17-Dec-2020].

[2] D. Monsters, “Text Preprocessing in Python: Steps, Tools, and Examples,” Medium, 15-Oct-2018. [Online]. Available: https://medium.com/@datamonsters/text-preprocessing-in-python-steps-tools-and-examples-bf025f872908. [Accessed: 17-Dec-2020].