CIS 660 Final Project(Fall 2022)

Sentiment analysis on amazon product reviews

By: Khatri Pooja(2783752)

Agenda

- Introduction
- Approach
- Dataset
- Cleaning and Preprocessing
- Analysis
- Preprocessing before building model
- Model building
- Test sentiments
- Challenges

Introduction

- Sentiment analysis is the process of detecting positive or negative sentiment in text.
- Polarity: Positive, Negative, Neutral
- Benefits:
  - Removes human bias through consistent analysis
  - Processes data at scale
  - Automation
  - Real-time analysis and insights

Approach

1. Rule-based Sentiment Analysis:
   - Step 1: “Lexicons” or lists of positive and negative words are created.
   - Step 2: Text processing
   - Step 3: A computer counts the number of positive or negative words in a particular text.
   - Step 4: The final step is to calculate the overall sentiment score for the text.
2. Machine Learning Sentiment Analysis:
   - Step 1: Feature Extraction
   - Step 2: Training & Prediction
   - Step 3: Classification algorithms
     - Naïve Bayes
     - Support Vector Machine
     - Maximum Entropy

Project Goal

Goal :

- Implement Naïve Bayes algorithm and Logistic regression on amazon reviews
- Analyze the sentiments

Platforms/System Tools Used: Sklearn library

Dataset

Dataset available at : https://jmcauley.ucsd.edu/data/amazon_v2/index.html

278677 rows × 9 columns

```
df.head()
```

	reviewerID	asin	reviewerName	helpful	reviewText	overall	summary	unixReviewTime	reviewTime
0	A1KLRMWW2FWPL4	0000031887	Amazon Customer "cameramom"	[0, 0]	This is a great tutu and at a really great pri...	5	Great tutu- not cheaply made	1297468800	02 12, 2011
1	A2G5TCU2WDFZ65	0000031887	Amazon Customer	[0, 0]	I bought this for my 4 yr old daughter for dan...	5	Very Cute!!	1358553600	01 19, 2013
2	A1RLQXYNCMWRWN	0000031887	Carola	[0, 0]	What can I say... my daughters have it in oran...	5	I have buy more than one	1357257600	01 4, 2013
3	A8U3FAMSJVHS5	0000031887	Caromcg	[0, 0]	We bought several tutus at once, and they are ...	5	Adorable, Sturdy	1398556800	04 27, 2014
4	A3GEOILWLK86XM	0000031887	CJ	[0, 0]	Thank you Halo Heaven great product for Little...	5	Grammy's Angels Love it	1394841600	03 15, 2014

Dataset Info

```
df.columns
```

```
Index(['reviewerID', 'asin', 'reviewerName', 'helpful', 'reviewText',
       'overall', 'summary', 'unixReviewTime', 'reviewTime'],
      dtype='object')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 278677 entries, 0 to 278676
Data columns (total 9 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   reviewerID      278677 non-null  object
 1   asin            278677 non-null  object
 2   reviewerName    278225 non-null  object
 3   helpful         278677 non-null  object
 4   reviewText      278677 non-null  object
 5   overall         278677 non-null  int64
 6   summary         278677 non-null  object
 7   unixReviewTime  278677 non-null  int64
 8   reviewTime      278677 non-null  object
dtypes: int64(2), object(7)
memory usage: 19.1+ MB
```

Description of columns in the file:

•reviewerID - ID of the reviewer, e.g. A2SUAM1J3GNN3B

•asin - ID of the product, e.g. 0000013714

•reviewerName - name of the reviewer

•helpful - helpfulness rating of the review, e.g. 2/3

•reviewText - text of the review

•overall - rating of the product

•summary - summary of the review

•unixReviewTime - time of the review (unix time)

•reviewTime - time of the review (raw)

Cleaning and Preprocessing

1. Handling NaN values

2. Create column reviews by concatenating review text and summary columns

3. Create sentiment column from based on overall rating from user

```
df['sentiment'].value_counts()

Positive    345845
Neutral      46358
Negative     44310
Name: sentiment, dtype: int64
```

4. Finding the helpfulness of the review: create helpful_rate feature which returns a/b value from [a,b]

5. Reviews column - Punctuation Cleaning and Removing Stop words

6. Remove unnecessary columns like reviewerName, unixReviewTime

```
df2['result'].value_counts()
```

```
0.00    314502
1.00     85683
0.50     10905
0.67      5083
0.75      3682
         ...
0.35         1
0.48         1
0.52         1
0.16         1
2.00         1
Name: result, Length: 95, dtype: int64
```

```
df['helpful_rate'] = df2['result']
```

```
df
```

	reviewerID	asin	reviewerName	overall	unixReviewTime	reviewTime	reviews	sentiment	helpful_rate
0	A1KLRMWW2FWPL4	0000031887	Amazon Customer "cameramom"	5	1297468800	02 12, 2011	This is a great tutu and at a really great pri...	Positive	0.0
1	A2G5TCU2WDFZ65	0000031887	Amazon Customer	5	1358553600	01 19, 2013	I bought this for my 4 yr old daughter for dan...	Positive	0.0
2	A1RLQXYNCMWRWN	0000031887	Carola	5	1357257600	01 4, 2013	What can I say... my daughters have it in oran...	Positive	0.0
3	A8U3FAMSJVHS5	0000031887	Caromcg	5	1398556800	04 27, 2014	We bought several tutus at once, and they are ...	Positive	0.0
4	A3GEOILWLK86XM	0000031887	CJ	5	1394841600	03 15, 2014	Thank you Halo Heaven great product for Little...	Positive	0.0
...	...	...	...	...	...	...	...	...	...
157831	A136YD08SCJ2LV	B00KMHKOZC	R. Spell "raspell"	5	1405296000	07 14, 2014	The Pet Magasin Retractable Dog Leash is the b...	Positive	0.0

```python
import re, string
def review_cleaning(text):
    '''Make text lowercase, remove text in square brackets,remove links,remove punctuation
    and remove words containing numbers.'''
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    return text
```

```python
df['reviews']=df['reviews'].apply(lambda x:review_cleaning(x))
df.head()
```

	reviewerID	asin	overall	reviewTime	reviews	sentiment	helpful_rate
0	A1KLRMWW2FWPL4	0000031887	5	02 12, 2011	this is a great tutu and at a really great pri...	Positive	0.0
1	A2G5TCU2WDFZ65	0000031887	5	01 19, 2013	i bought this for my yr old daughter for danc...	Positive	0.0
2	A1RLQXYNCMWRWN	0000031887	5	01 4, 2013	what can i say my daughters have it in orange ...	Positive	0.0
3	A8U3FAMSJVHS5	0000031887	5	04 27, 2014	we bought several tutus at once and they are g...	Positive	0.0
4	A3GEOILWLK86XM	0000031887	5	03 15, 2014	thank you halo heaven great product for little...	Positive	0.0

```python
stop_words= ['yourselves', 'between', 'whom', 'itself', 'is', "she's", 'up', 'herself', 'here', 'your', 'each',
             'we', 'he', 'my', "you've", 'having', 'in', 'both', 'for', 'themselves', 'are', 'them', 'other',
             'and', 'an', 'during', 'their', 'can', 'yourself', 'she', 'until', 'so', 'these', 'ours', 'above',
             'what', 'while', 'have', 're', 'more', 'only', "needn't", 'when', 'just', 'that', 'were', "don't",
             'very', 'should', 'any', 'y', 'isn', 'who',  'a', 'they', 'to', 'too', "should've", 'has', 'before',
             'into', 'yours', "it's", 'do', 'against', 'on',  'now', 'her', 've', 'd', 'by', 'am', 'from',
             'about', 'further', "that'll", "you'd", 'you', 'as', 'how', 'been', 'the', 'or', 'doing', 'such',
             'his', 'himself', 'ourselves',  'was', 'through', 'out', 'below', 'own', 'myself', 'theirs',
             'me', 'why', 'once',  'him', 'than', 'be', 'most', "you'll", 'same', 'some', 'with', 'few', 'it',
             'at', 'after', 'its', 'which', 'there','our', 'this', 'hers', 'being', 'did', 'of', 'had', 'under',
             'over','again', 'where', 'those', 'then', "you're", 'i', 'because', 'does', 'all']
```

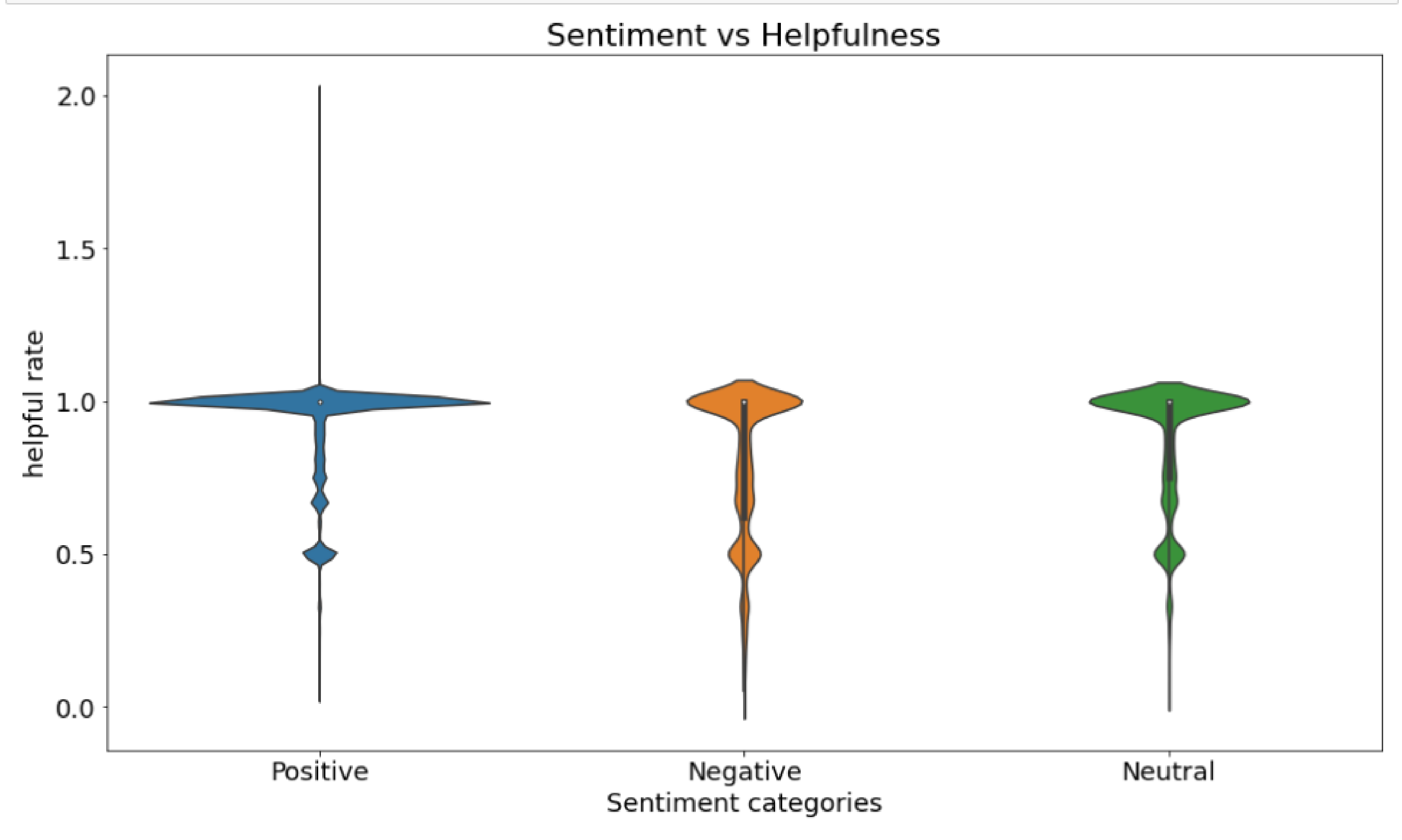
```python
df['reviews'] = df['reviews'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop_words)]))
```

	reviewerID	asin	overall	reviewTime	reviews	sentiment	helpful_rate
0	A1KLRMWW2FWPL4	0000031887	5	02 12, 2011	great tutu really great price doesnt look chea...	Positive	0.0
1	A2G5TCU2WDFZ65	0000031887	5	01 19, 2013	bought yr old daughter dance class wore today ...	Positive	0.0
2	A1RLQXYNCMWRWN	0000031887	5	01 4, 2013	say daughters orange black white pink thinking...	Positive	0.0
3	A8U3FAMSJVHS5	0000031887	5	04 27, 2014	bought several tutus got high reviews sturdy s...	Positive	0.0
4	A3GEOILWLK86XM	0000031887	5	03 15, 2014	thank halo heaven great product little girls g...	Positive	0.0

Sentiments vs Helpful rate

```
pd.DataFrame(df.groupby('sentiment')['helpful_rate'].mean())
```

	helpful_rate
sentiment	
Negative	0.318486
Neutral	0.250437
Positive	0.242943



Insight: more number of positive reviews are having high helpful rate

Creating few more features for text analysis

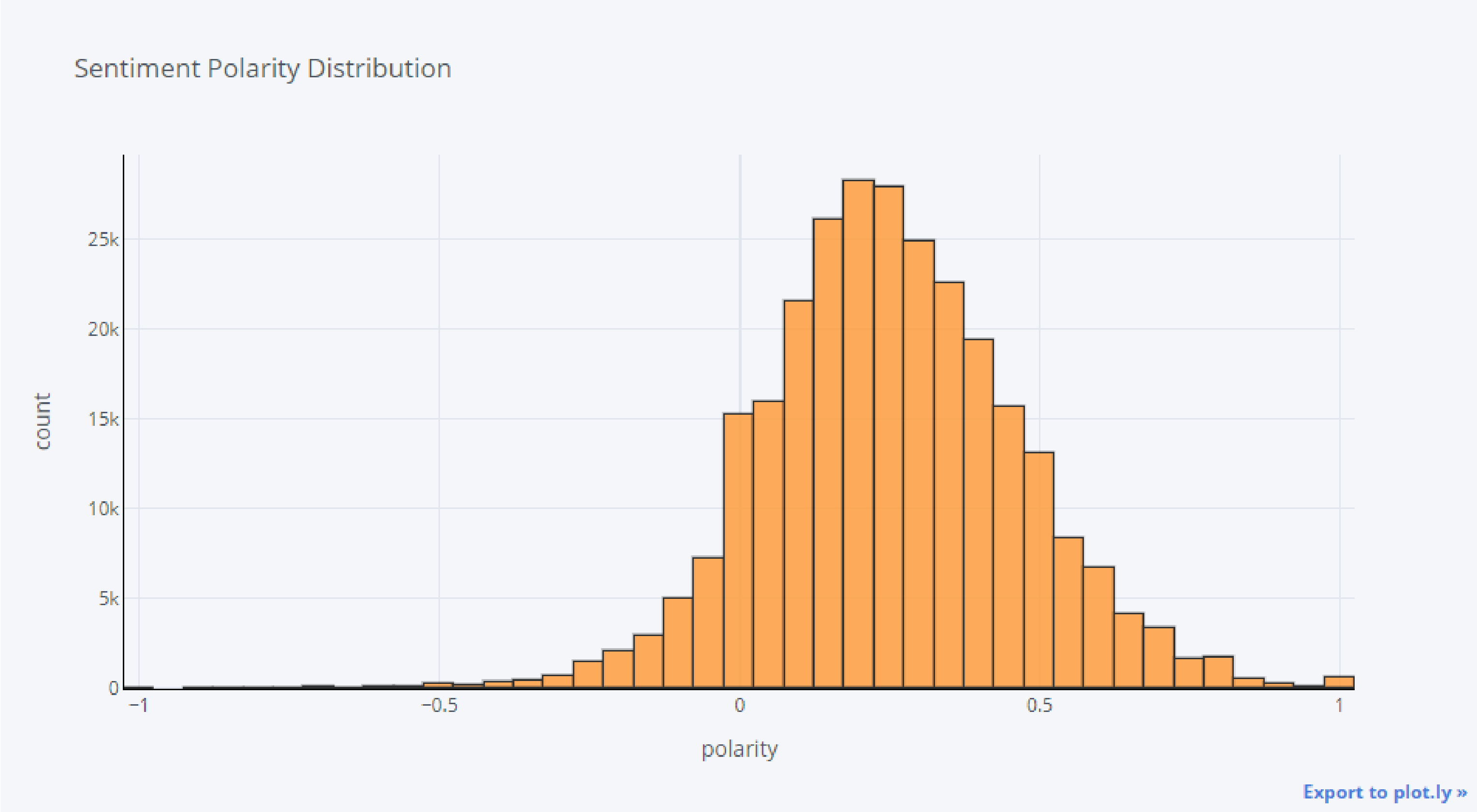
Now, let's create polarity, review length and word count

Polarity: We use Textblob for figuring out the rate of sentiment . It is between [-1,1] where -1 is negative and 1 is positive polarity

Review length: length of the review which includes each letters and spaces

Word length: This measures how many words are there in review

Now review the polarity distribution and Review Rating Distribution



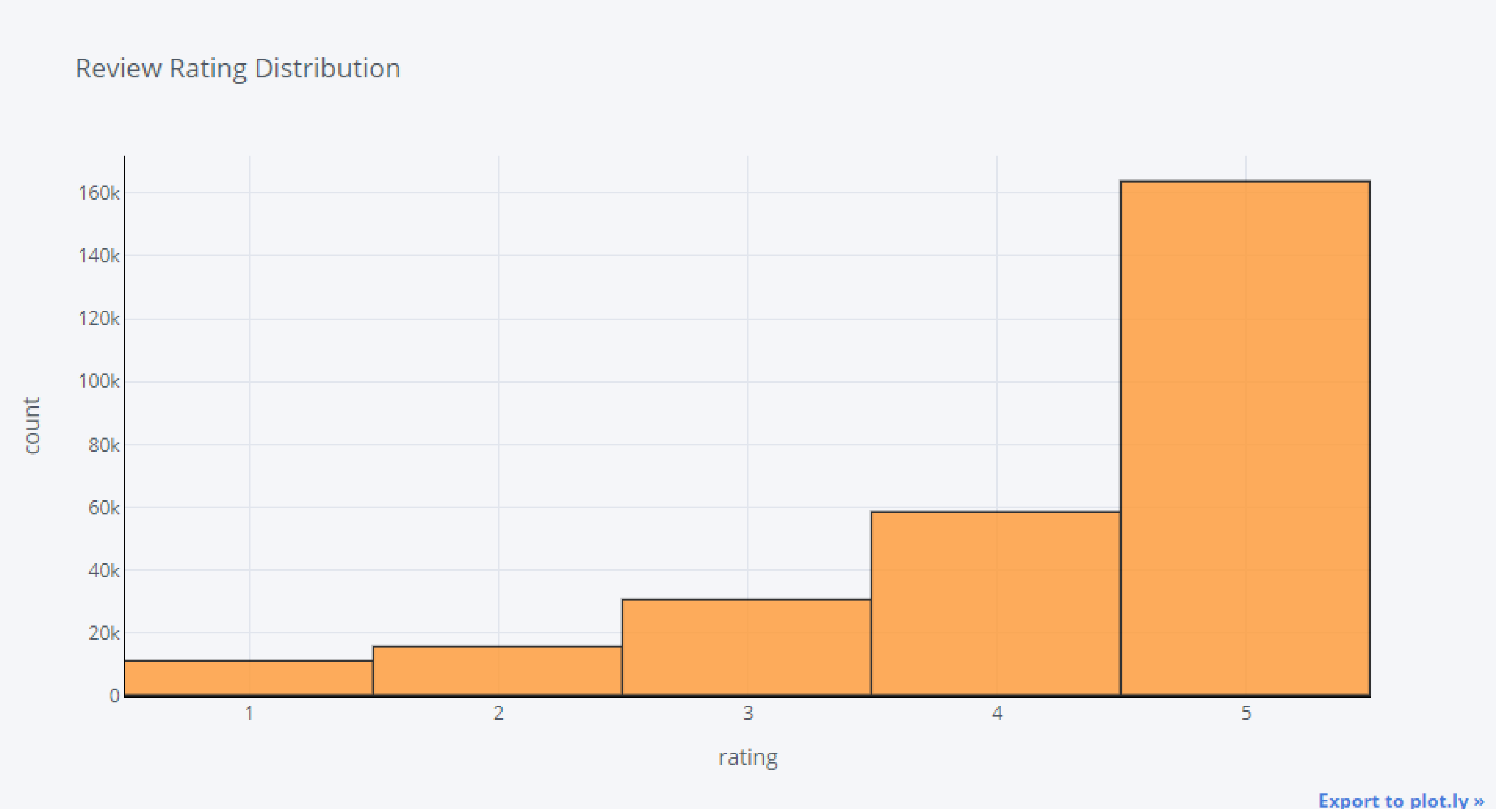
Insights:

We have a lot of positive polarities compared to the negative polarities

This polarity distributions assures the number of positive reviews we had

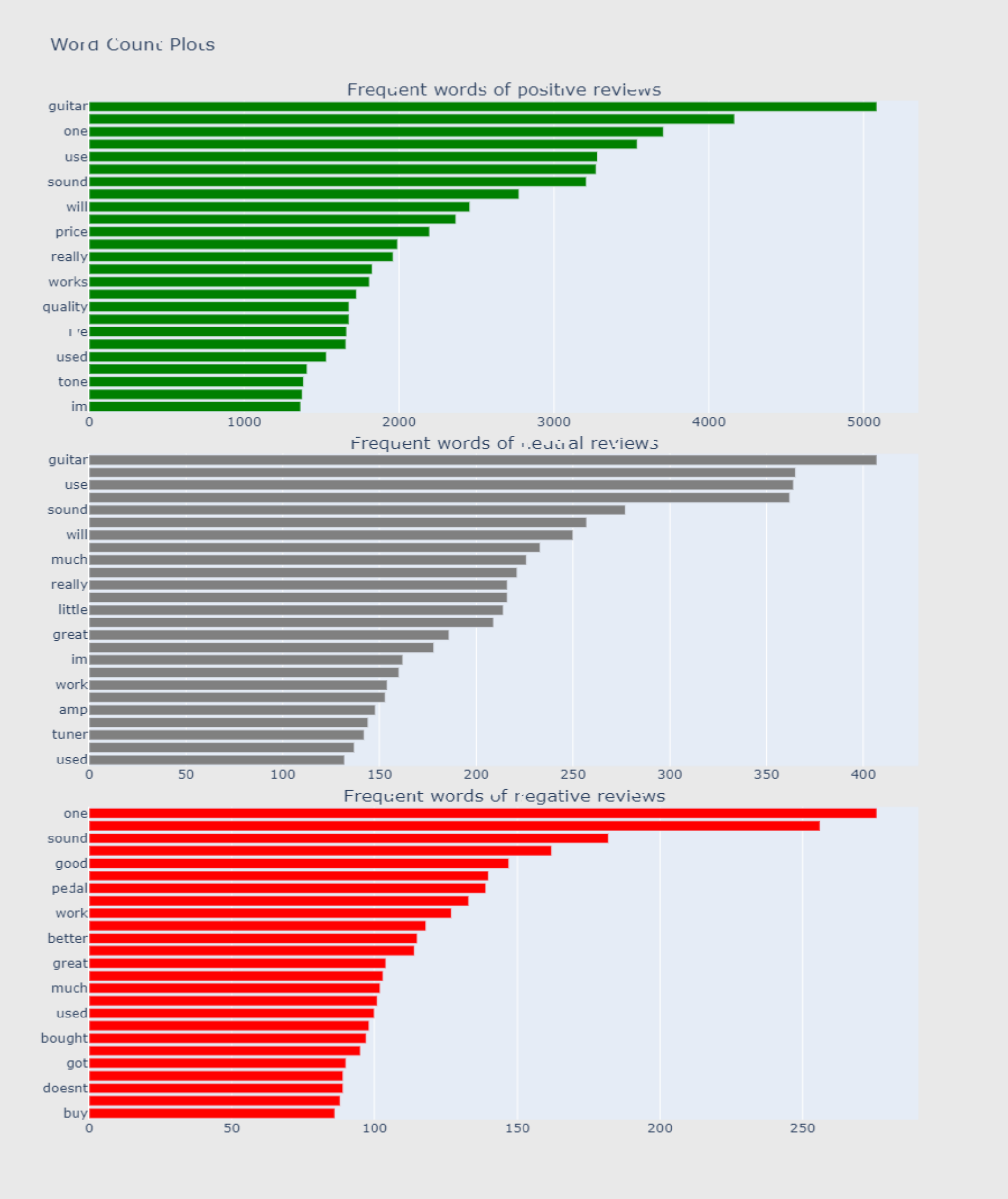
We can say that this polarity is a normally distributed but not standard normal

```
df['overall'].iplot(
    kind='hist',
    xTitle='rating',
    linecolor='black',
    yTitle='count',
    title='Review Rating Distribution')
```



Insight: We have a large number of 5 ratings(nearly 160k) followed by 4,3,2,1. It's linear in nature

As we see, the words doen't match with the sentiment except few



Before building model..

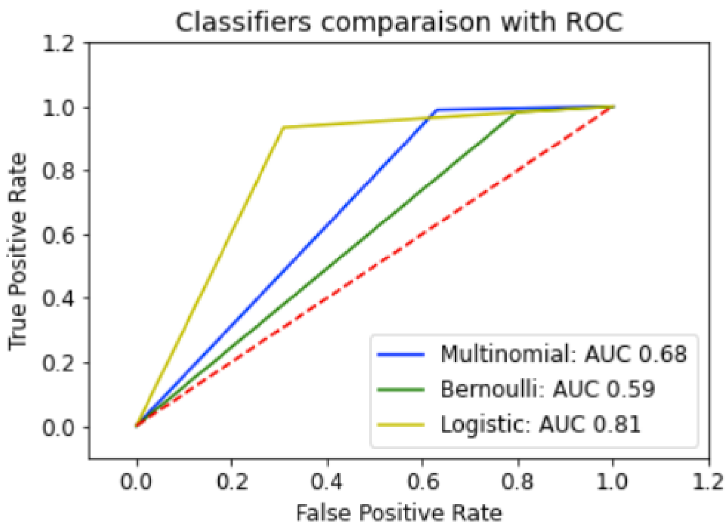
- Word Cloud
- Extracting Features from Cleaned reviews
- Encoding target variable-sentiment
- Stemming the reviews
- TFIDF(Term Frequency — Inverse Document Frequency)
- Handling Imbalance target feature-SMOTE
- Train-test split(75:25)

Model building

Naïve Bayes :

- Multinomial
- Bernoulli

Logistic Regression




```
accuracy_score(y_test, prediction['Logistic'])
```

0.8840067460887039

```
accuracy_score(y_test, prediction['Bernoulli'])
```

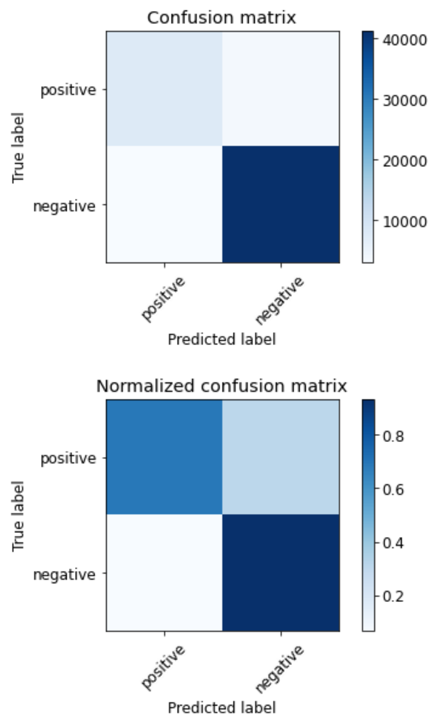
0.8215336586766183

```
accuracy_score(y_test, prediction['Multinomial'])
```

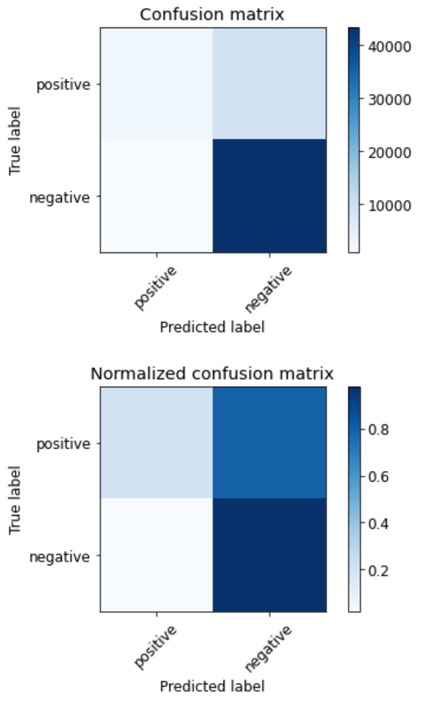
0.8615257643174967

Confusion Matrix

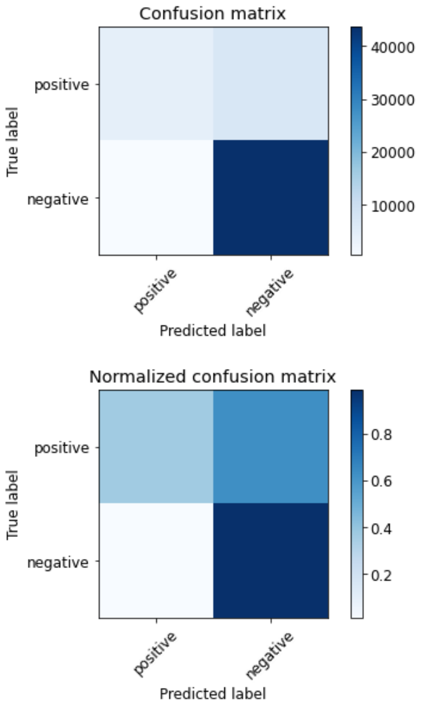
Logistic



Bernoulli



Multinomial



```python
print(metrics.classification_report(y_test, prediction['Logistic'], target_names = ["positive", "negative"]))
```

	precision	recall	f1-score	support
positive	0.73	0.69	0.71	11457
negative	0.92	0.93	0.93	44279
accuracy			0.88	55736
macro avg	0.83	0.81	0.82	55736
weighted avg	0.88	0.88	0.88	55736

```python
accuracy_score(y_test, prediction['Logistic'])
```

```python
print(metrics.classification_report(y_test, prediction['Bernoulli'], target_names = ["positive", "negative"]))
```

	precision	recall	f1-score	support
positive	0.74	0.20	0.32	11457
negative	0.83	0.98	0.90	44279
accuracy			0.82	55736
macro avg	0.78	0.59	0.61	55736
weighted avg	0.81	0.82	0.78	55736

```python
print(metrics.classification_report(y_test, prediction['Multinomial'], target_names = ["positive", "negative"]))
```

	precision	recall	f1-score	support
positive	0.90	0.37	0.52	11457
negative	0.86	0.99	0.92	44279
accuracy			0.86	55736
macro avg	0.88	0.68	0.72	55736
weighted avg	0.87	0.86	0.84	55736

Test sentiments

```
def testSentiments(model, testData):
    testCounts = countVector.transform([testData])
    testTfidf = tfidf_transformer.transform(testCounts)
    result = model.predict(testTfidf)[0]
    probability = model.predict_proba(testTfidf)[0]
    print("Sample estimated as %s: negative prob %f, positive prob %f" % (result.upper(), probability[0], probability[1]))

testSentiments(logreg, "Heavenly Highway Hymns")
testSentiments(logreg, "Very oily and creamy. Not at all what I expected... ordered this to try to highlight and contour and
testSentiments(logreg, "Shampoo smells so good!")
```

```
Sample estimated as POSITIVE: negative prob 0.000328, positive prob 0.999672
Sample estimated as NEGATIVE: negative prob 0.999999, positive prob 0.000001
Sample estimated as POSITIVE: negative prob 0.141032, positive prob 0.858968
```

Conclusion

- Consider welcoming ngram in sentiment analysis as one word can't give is proper results and stop words got to be manually checked as they have negative words. It is advised to avoid using stop words in sentiment analysis
- Balancing the dataset gives good accuracy score. Without balancing, I got good precision, but very bad recall and in turn affected my f1 score. So, balancing the target feature is important

Sentiment Analysis Challenges

- Subjectivity
- Emojis
- Idioms
- Neutrality

References

- https://getthematic.com/sentiment-analysis/#:~:text=Sentiment%20analysis%20uses%20machine%20learning,based%20and%20automated%20sentiment%20analysis
- https://mickzhang.com/amazon-reviews-using-sentiment-analysis
- chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/http://www.narimanfarsad.com/cps803/docs/samples/CPS803-SampleReport-SentimentAnalysis.pdf
- https://scholarworks.rit.edu/cgi/viewcontent.cgi?article=12196&context=theses

Thank You!