REVIEW RATING PREDICTION PROJECT

Submitted by:

KONATALA MOHIT

ACKNOWLEDGMENT

I am grateful for the writer, developers and authority of the website pages geekforgeeks, wikipedia and stackoverflow which have helped me to refer for whenever there was need for guidance. Feature Engineering and usage of NLP helped me to get a better data to analyse and explore data. I am mostly grateful to DataTrained and Jose Portilla who have guided me in learning and enhancing my data science and NLP skills required to analyse and solve this project.

INTRODUCTION:

In this report I will be discussing about text classification of full review of an electronic product by using nltk library and few of the machine learning models via python and its libraries. On E-commerce platform reviews and rating are the important annotations which define the sales of a product also help the back-end sales executive and distributors to get handful feedback on the product which in return helps to improvise the product. This project to help a client who has a website where different set of audience write a review for the product. And to cut short the review into a review we need to a model which can rate the review in star rating scaling from 1 to 5. To rate a full review of a product, one must have a good understanding of natural language processing, and usage of natural language toolkit to make the required data abled to feed it to a machine learning model.

Review of Literature:

For the better understanding of the data refer to Natural Language Processing & should know set of language and abusive words used in the data. Feature engineering is used to help in inspection of the comments before labelling it. Usage of Pipeline is required to simplify the process and required step by step process to clean and pre-process the data.

Undertaken Problem:

Objective of the project is to predict the rating of the full text review i.e., how much rating can the review of a customer weighs i.e., by analysing various factors that affect the rating outcome. Label prediction is one the many problems in Data Science as it ticks all the basic fundamentals required to analyse data and creating a machine learning model to predict the necessary outcome.

Mathematical/ Analytical Modelling of the Problem:

Rating is the target data in the dataset and the variables present in the target data are in binary form, therefore Classification Machine Learning Models are used in this project. Classification is for estimating categorical form of data by using established relation between feature and target variables. Visual Analysis is used to study and observe the data. This prediction has multi-class classification.

Data Sources and their formats:

The data is extracted from flipkart website using selenium library. The data consists of various reviews and ratings of electronic products. The data is provided in the excel format. Train data contains 28005 entries having 6 variables in the dataset.

Data Pre-processing:

Data pre-processing has two main steps i.e., Data Cleaning and Data Transforming.

Data Cleaning:

Data Cleaning is one of the most important steps creating a machine model. If an uncleaned data is fed to a machine learning model, then the model will perform very poorly.

The Dataset contains four null values in it. As the count of null values, we can drop those columns. Unnamed: 0 and 0.1 (serial number) is present in the dataset as it unnecessary hence dropping these two columns. Review text is in object string format.

Text Full Review Data has enormous junk words in it which aren't useful in our label predict, therefore one must remove all the junk and unnecessary words present in the full review text column.

Cleaning of the text data is as follows:

1. Word Tokenization.
2. Removal of punctuations.
3. Removal of stopwords.
4. Lemmatization.

Word Tokenization: with the help of nltk library we convert the string comment text into words so that is easier for processing to initiate. word_tokenize is the keyword used to convert a string into words.

Removal of punctuations: In English language usage of punctuation is common in order sound grammatically correct but in machine learning punctuation are not necessary so we remove the punctuation from the text. From string library, we call punctuation in order to remove the punctuations from the comments.

Removal of stop words: Same as the punctuations stop words are not necessary so we remove them by using stopwords keyword. This keyword already has all the commonly used stop words in our day-to-day life, although if one wants to add few more punctuation one can add those to this stopwords by modifying it. Punctuation and stop words are majority of the junk present in a text.

Lemmatization: It is process to simplify the verb form in order to decrease the uniqueness of words in the comments lot.

After performing these above stepped we join the words and make it to string format as before and add in to a list which is later converted into data frame and append to our existing data frame. Now we name it as clean comment text.

Note: Label column is slightly imbalanced.

Data Transforming:

After Data Cleaning the data must be transformed into numerical form as one can't feed ordinal data to machine learning model and also the data must be normalized as normalizing the data the machine learning model gives equal importance to all the data.

In this project I am using label token frequency inverse document frequency vectorizer to transform and fit the ordinal data present in the data set into numerical form. Tfidfvectorizer not convert the data into numerical form but has the ability to normalize the data. Before standardizing the data, we must split the data into train and test sets. Train Test Split method is to split the data into train and test data

After separating the train and test data we use Pipeline for model training and prediction as it makes the process simpler.

Data Inputs- Logic- Output Relationships:

After Separating the Input (feature) and output (target) data we split them into train and test division one part of the data is used to train the ML model and other part of the data to predict the output. When the train data is fed to machine learning model it generates an algorithm or simply put an equation that is applicable to all the data and when test data is fed to it implements the trained data equation to the current input values to predict the outcome. If the input has no outliers and is clean that there is no over or underfitting in the outcome.

Hardware and Software Requirements and Tools Used:

Hardware Required for Jupyter Notebook Software is as follows:

Memory and disk space required per user: 1GB RAM + 1GB of disk + .5 CPU core.

Server overhead: 2-4GB or 10% system overhead (whatever is larger), .5 CPU cores, 10GB disk space.

Port requirements: Port 8000 plus 5 unique, random ports per notebook

Libraries Used:

Pandas library: To frame raw data, visualize and perform task on it via other libraries.

Numpy library: To perform mathematical functions on the framed data numpy is used. In this project used it to location nan values and replace them with desired value also to find mean and standard values.

Matplotlib library: This library is used to visualize the data. Used to visualize univariate and bivariate analysis (pie plot, count plot and scatter plot) also to visualize outliers via box plot.

Seaborn library: heatmap to see co-relation between feature variable to arrest high collinearity.

Sklearn library: Imported this library to normalize the data, split the data into train and test data, various machine learning model and cross validation techniques.

Warnings library: To ignore filter warning shown while compiling block of codes

Pickle: To save the trained machine learning model.

Nltk: Imported this library to clean and preprocess the text data.

Algorithm Used:

1. Naïve Bayes MultinomialNB
2. Random Forest Classifier
3. Logistic Regression
4. Linear SVC

Model Training and Selection:

Naïve Bayes Multinomial NB:

The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as tf-idf may also work

```
text_clf_4=Pipeline(steps=[('tfidfvectorizer',
                 TfidfVectorizer(max_df=0.75, ngram_range=(1, 3))),
                ('multinomialnb', MultinomialNB(alpha=0.01))])
text_clf_4.fit(X_train,y_train)
p4=text_clf_4.predict(X_test)
print(classification_report(p4, y_test))
```

```
              precision    recall  f1-score   support

           0       0.70      0.64      0.67      1328
           1       0.38      0.54      0.44       648
           2       0.37      0.42      0.39      1074
           3       0.53      0.45      0.49      1414
           4       0.61      0.56      0.59      1137

    accuracy                           0.52      5601
   macro avg       0.52      0.52      0.52      5601
weighted avg       0.54      0.52      0.53      5601
```

Random Forest Classifier:

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting

```
text_clf_3=Pipeline(steps=[('tfidfvectorizer',
           TfidfVectorizer(max_df=0.75, ngram_range=(1, 3))),
           ('randomforestclassifier', RandomForestClassifier(n_estimators=350))])
text_clf_3.fit(X_train,y_train)
p3=text_clf_3.predict(X_test)
print(classification_report(p3, y_test))
```

```
              precision    recall  f1-score   support

           0       0.77      0.60      0.68      1550
           1       0.41      0.54      0.46       698
           2       0.35      0.46      0.40       925
           3       0.59      0.47      0.52      1512
           4       0.55      0.62      0.58       916

    accuracy                           0.54      5601
   macro avg       0.53      0.54      0.53      5601
weighted avg       0.57      0.54      0.55      5601
```

Logistic Regression:

Logistic regression is a process of modelling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome, something that can take two values such as true/false, yes/no, and so on.

```
text_clf_2=Pipeline(steps=[('tfidfvectorizer',
                 TfidfVectorizer(max_df=0.75, ngram_range=(1, 3))),
                ('logisticregression', LogisticRegression(solver='newton-cg'))])
text_clf_2.fit(X_train,y_train)
p2=text_clf_2.predict(X_test)
print(classification_report(p2, y_test))
```

	precision	recall	f1-score	support
0	0.77	0.63	0.69	1495
1	0.40	0.57	0.47	649
2	0.37	0.46	0.41	967
3	0.60	0.47	0.53	1534
4	0.56	0.60	0.58	956
accuracy			0.55	5601
macro avg	0.54	0.55	0.54	5601
weighted avg	0.58	0.55	0.55	5601

Linear Support Vector Classification:

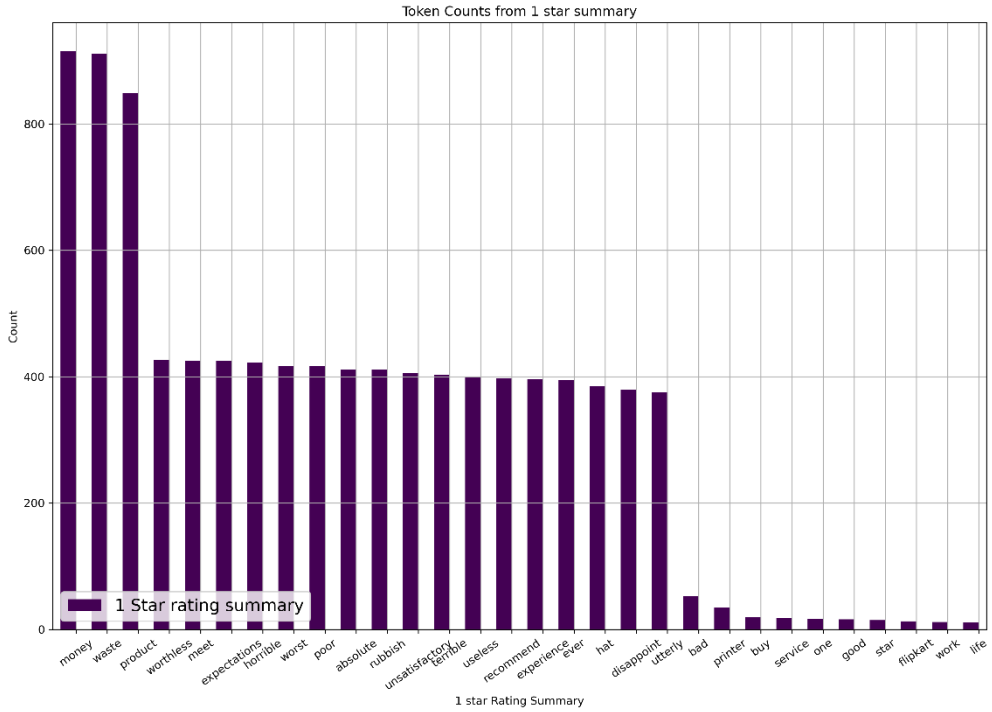
Similar to SVC with parameter kernel=’linear’, but implemented in terms of liblinear rather than libsvm, so it has more flexibility in the choice of penalties and loss function and should scale better to large number of samples.

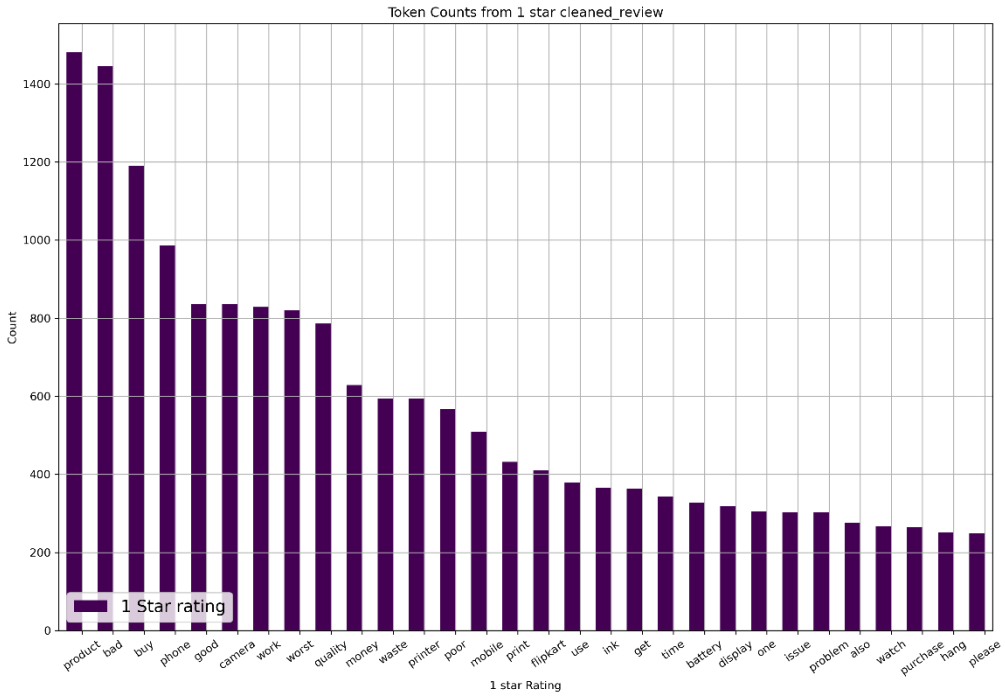
This class support both dense and sparse input and the multiclass support is handled according to a one-vs-the-rest scheme.

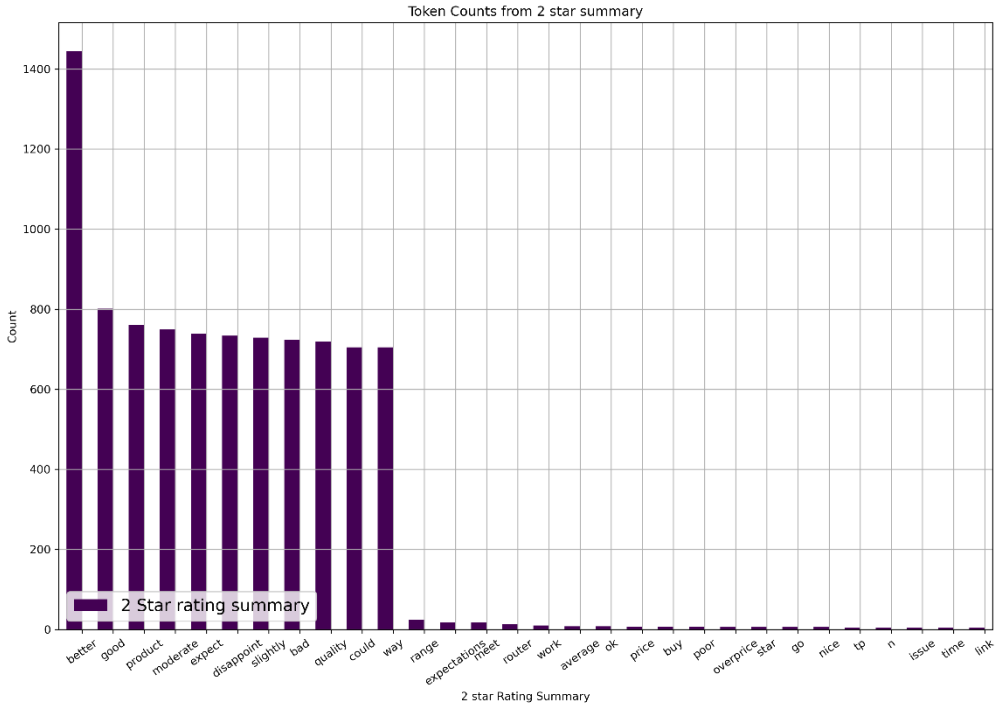
```
text_clf_1= Pipeline(steps=[('tfidfvectorizer',
                 TfidfVectorizer(max_df=0.75, ngram_range=(1, 3))),
                ('linearsvc', LinearSVC(C=0.1))])
text_clf_1.fit(X_train,y_train)
p1=text_clf_1.predict(X_test)
print(classification_report(p1, y_test))
```

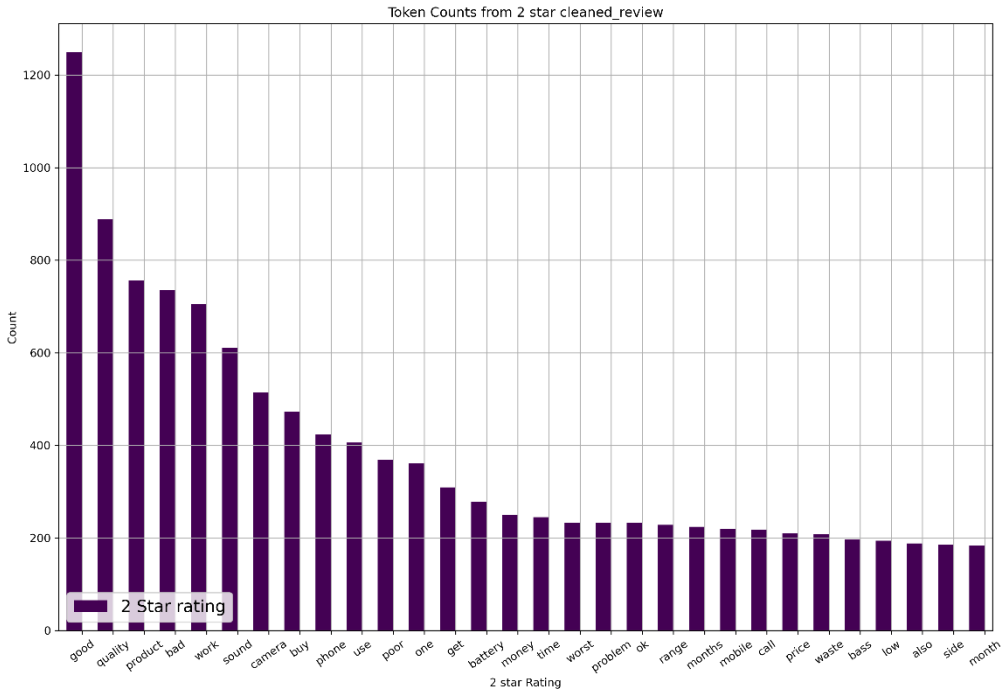
	precision	recall	f1-score	support
0	0.80	0.60	0.69	1631
1	0.38	0.59	0.46	591
2	0.33	0.49	0.40	830
3	0.59	0.47	0.52	1523
4	0.59	0.59	0.59	1026
accuracy			0.54	5601
macro avg	0.54	0.55	0.53	5601
weighted avg	0.59	0.54	0.56	5601

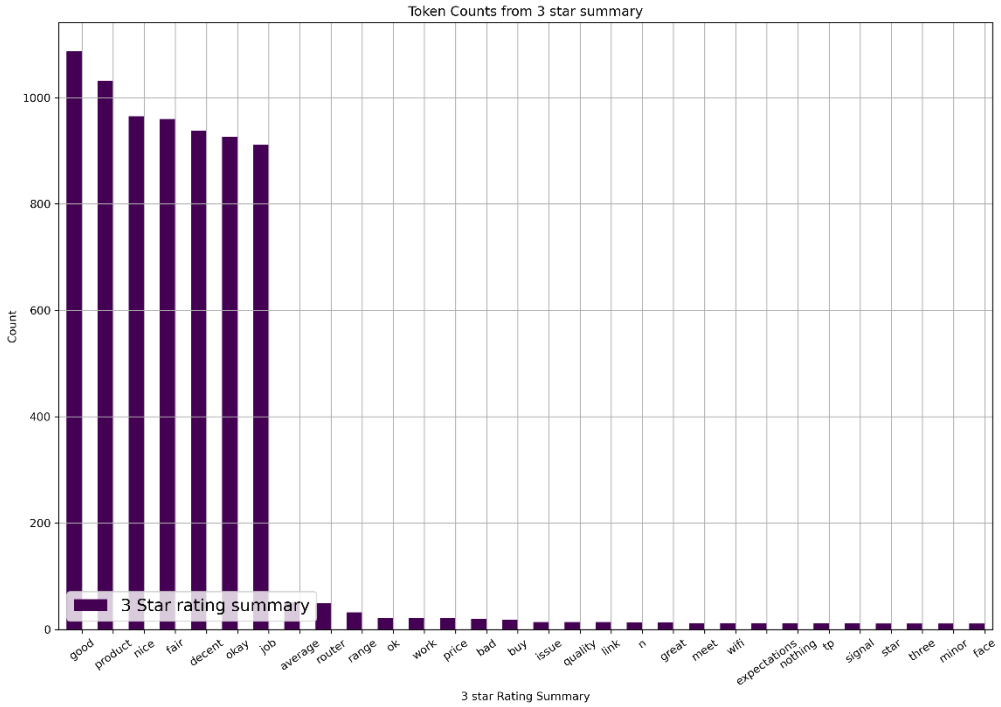
Visualization:

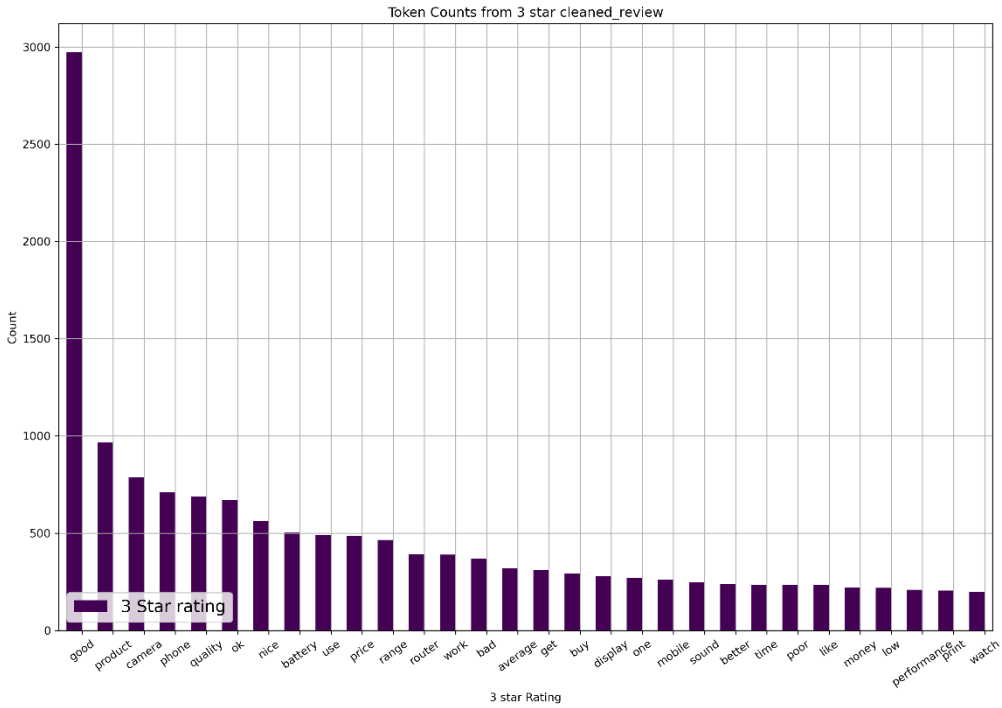


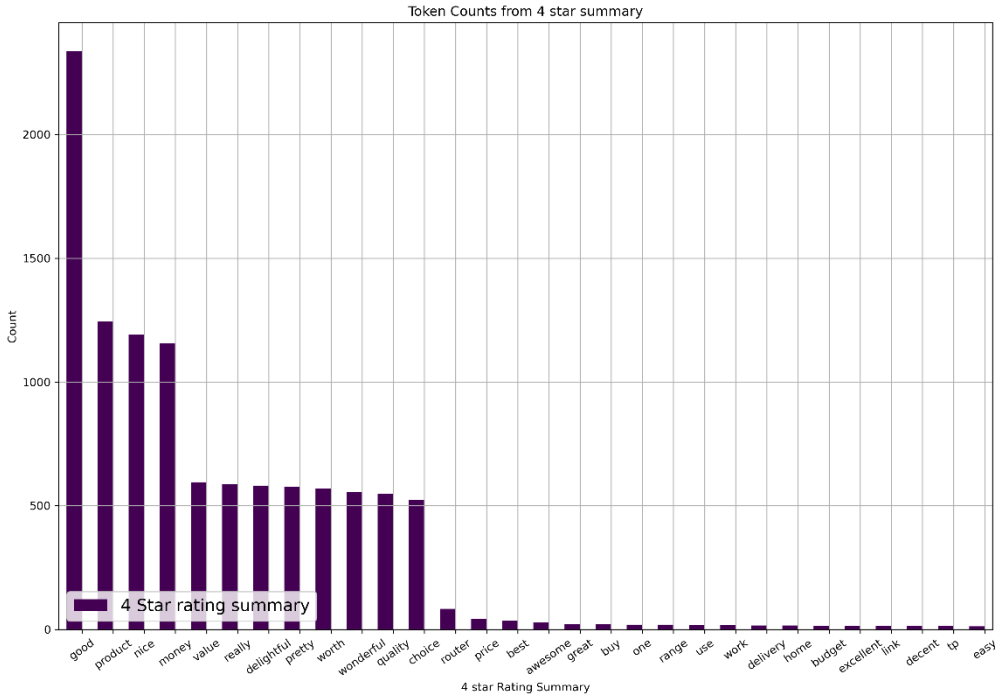


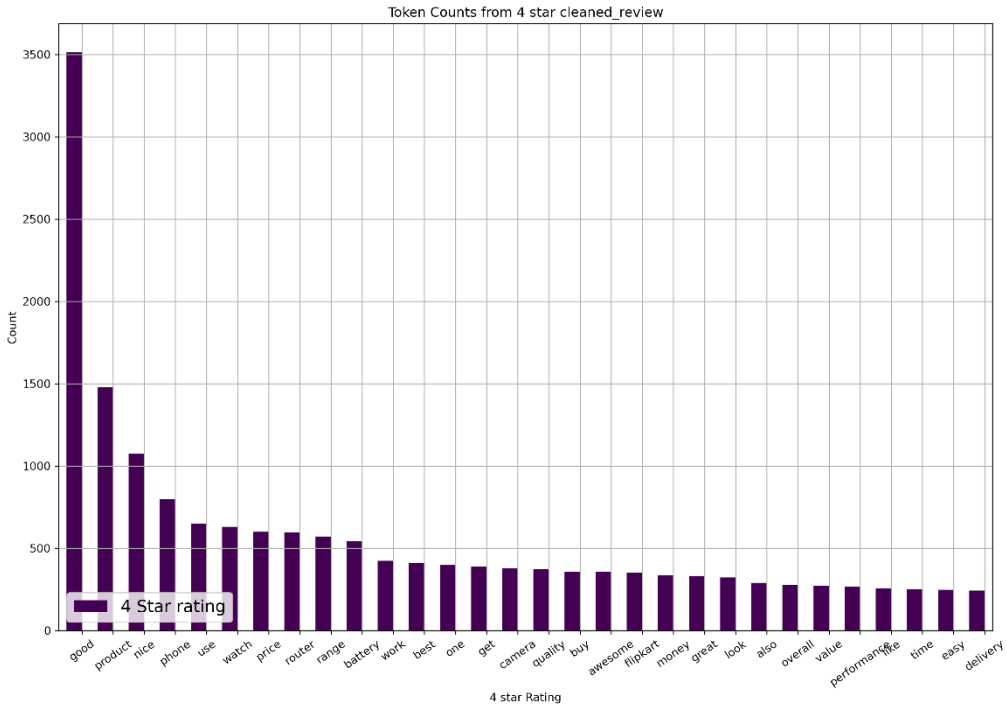

Token Counts from 2 star summary
Count
2 Star rating summary
better
good
product
moderate
expect
disappoint
slightly
bad
quality
could
way
range
expectations
meet
router
work
average
ok
price
buy
poor
overprice
star
go
nice
tp
n
issue
time
link
2 star Rating Summary

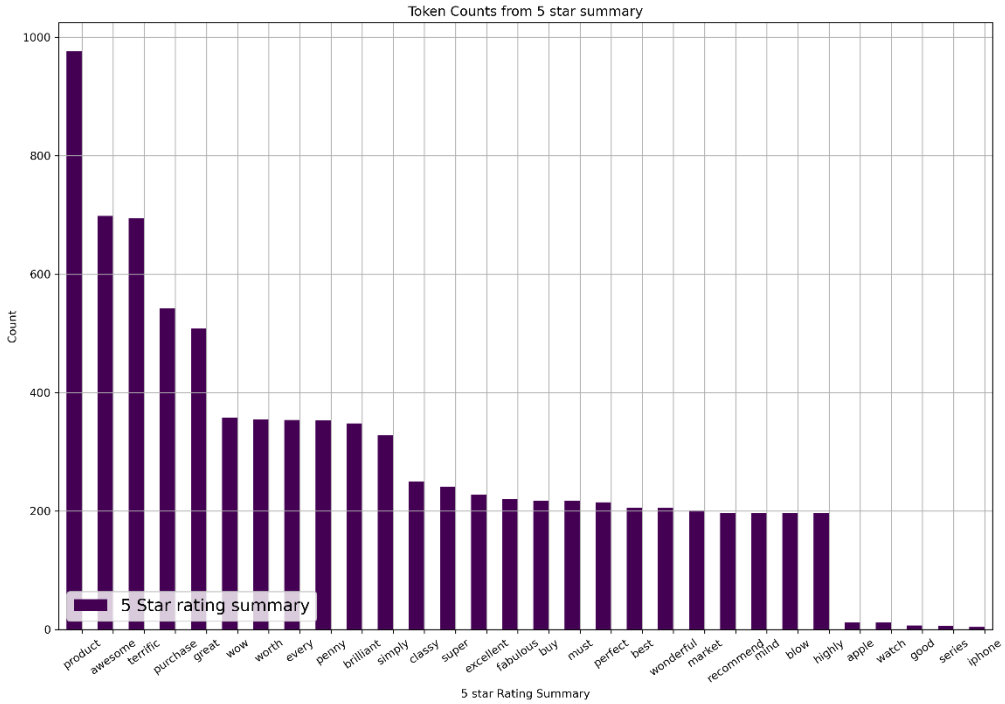

Token Counts from 2 star cleaned_review
Count
2 Star rating
good
quality
product
bad
work
sound
camera
buy
phone
use
poor
one
get
battery
money
time
worst
problem
ok
range
months
mobile
call
price
waste
bass
low
also
side
month
2 star Rating

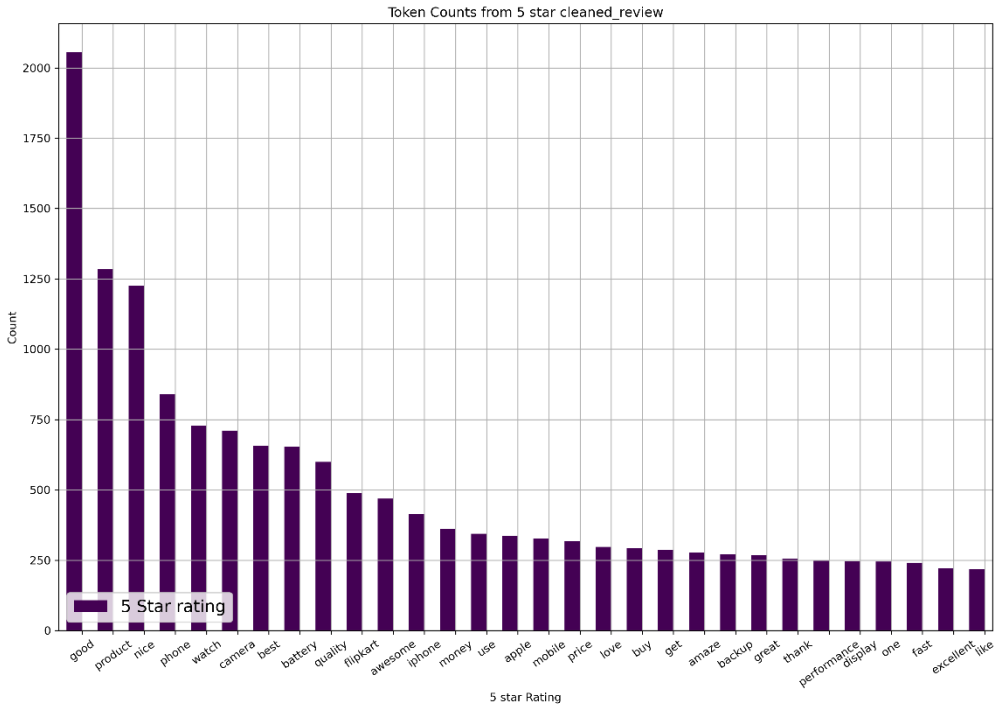

Token Counts from 3 star summary
Count
3 Star rating summary
good
product
nice
fair
decent
okay
job
average
router
range
ok
work
price
bad
buy
issue
quality
link
n
great
meet
wifi
expectations
nothing
tp
signal
star
three
minor
face
3 star Rating Summary

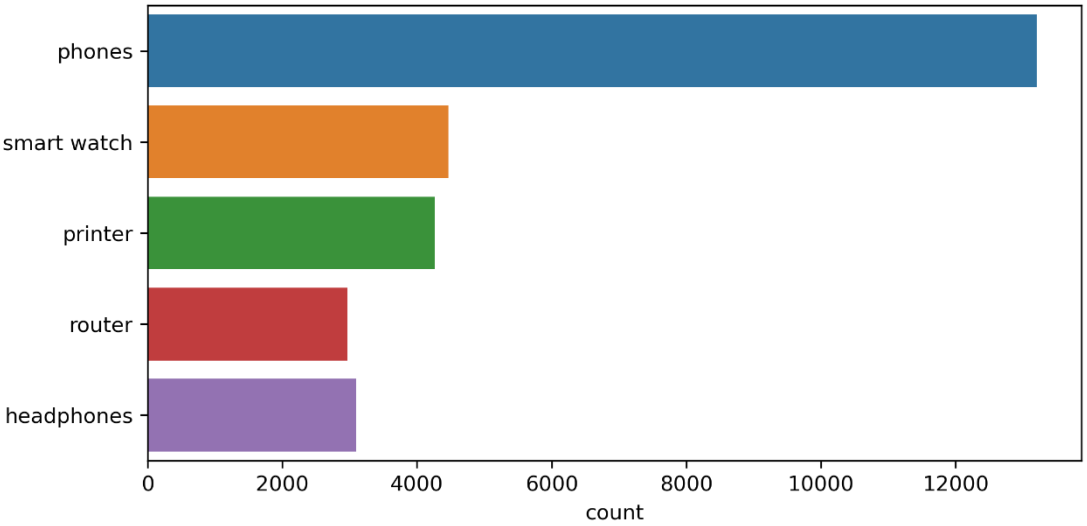

Token Counts from 3 star cleaned_review
Count
3 Star rating
good
product
camera
phone
quality
ok
nice
battery
use
price
range
router
work
bad
average
get
buy
display
one
mobile
sound
better
time
poor
like
money
low
performance
print
watch
3 star Rating

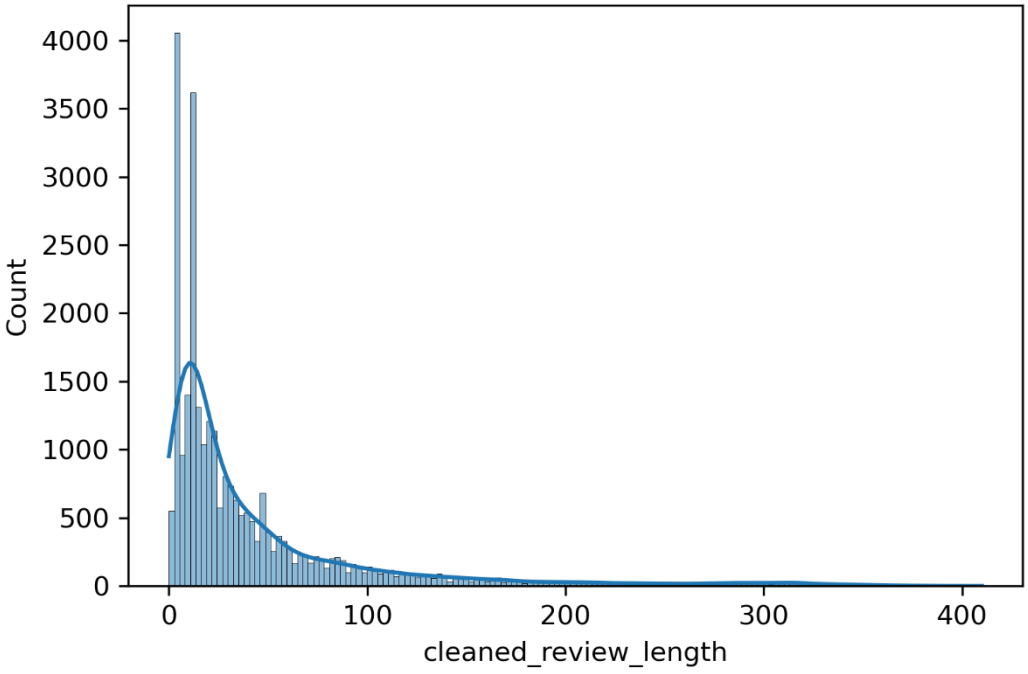

Token Counts from 4 star summary
Count
4 Star rating summary
good
product
nice
money
value
really
delightful
pretty
worth
wonderful
quality
choice
router
price
best
awesome
great
buy
one
range
use
work
delivery
home
budget
excellent
link
decent
tp
easy
4 star Rating Summary

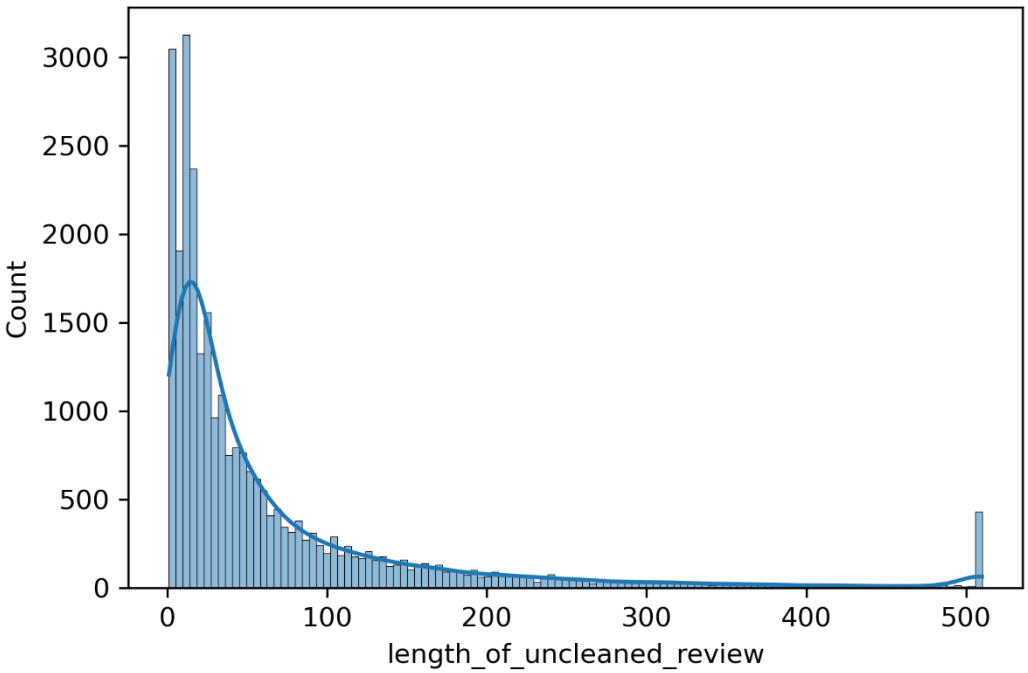

Token Counts from 4 star cleaned_review
Count
4 Star rating
good
product
nice
phone
use
watch
price
router
range
battery
work
best
one
get
camera
quality
buy
awesome
flipkart
money
great
look
also
overall
value
performance
like
time
easy
delivery
4 star Rating

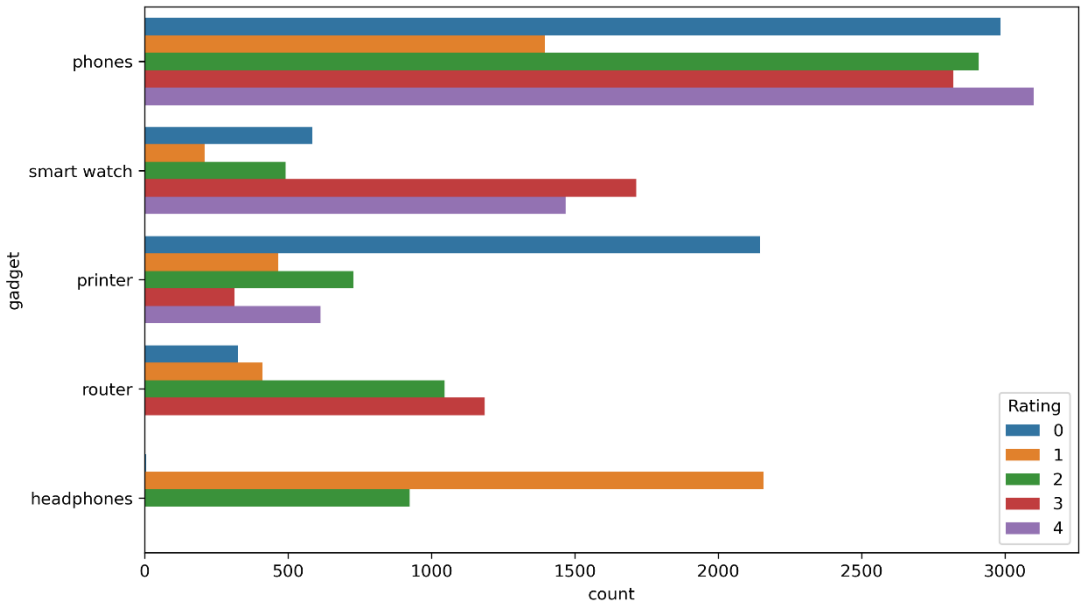

Token Counts from 5 star summary
Count
5 Star rating summary
product
awesome
terrific
purchase
great
wow
worth
every
penny
brilliant
simply
classy
super
excellent
fabulous
buy
must
perfect
best
wonderful
market
recommend
mind
blow
highly
apple
watch
good
series
iphone
5 star Rating Summary

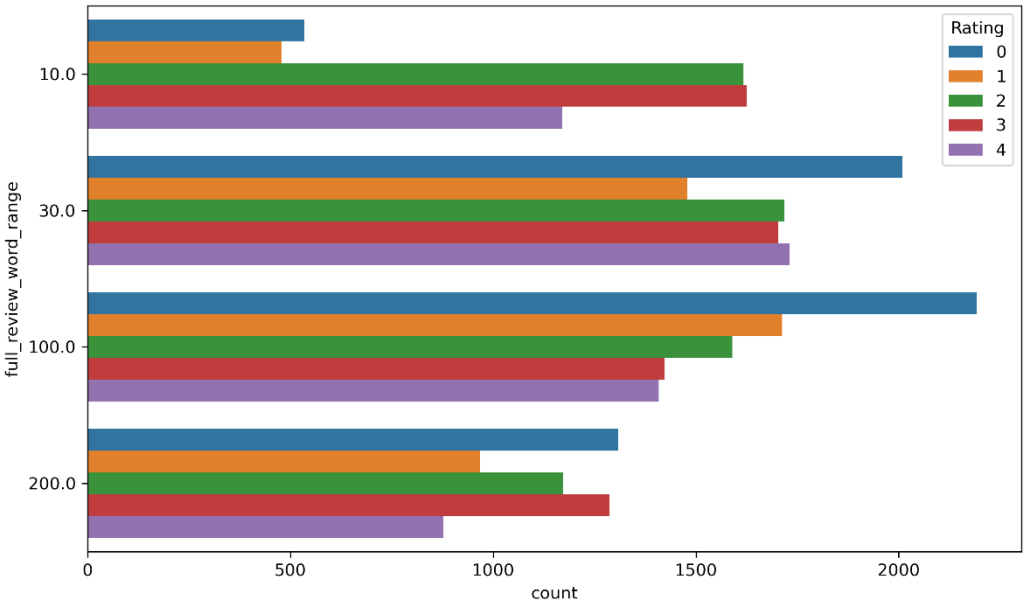

Token Counts from 5 star cleaned_review
Count
5 Star rating
good
product
nice
phone
watch
camera
best
battery
quality
flipkart
awesome
iphone
money
use
apple
mobile
price
love
buy
get
amaze
backup
great
thank
performance
display
one
fast
excellent
like
5 star Rating

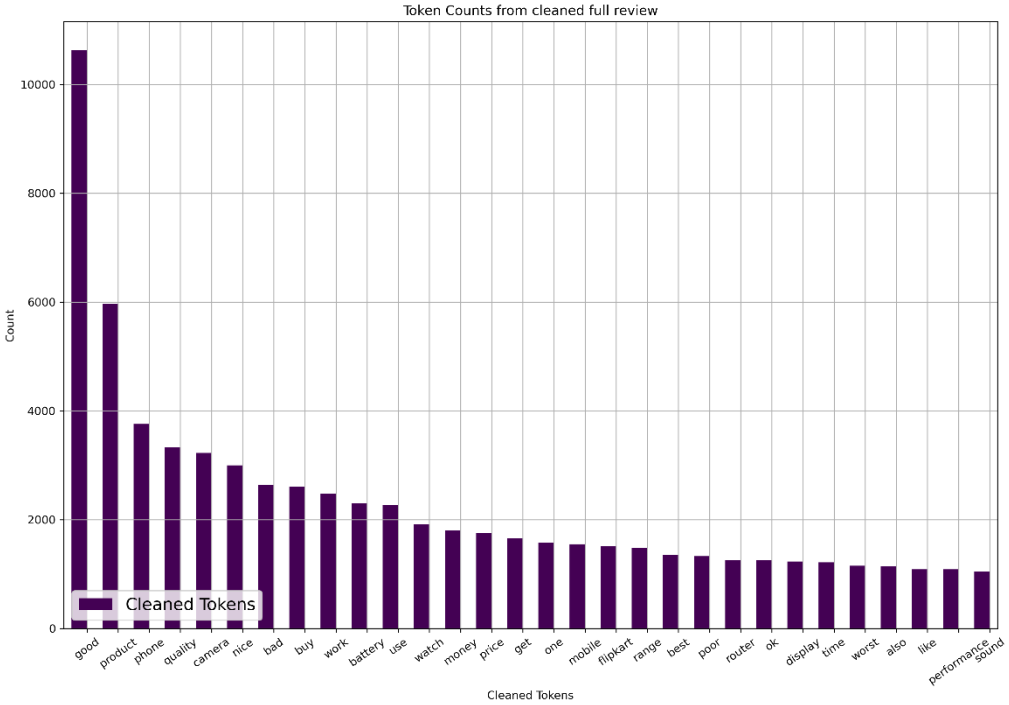

phones
smart watch
printer
router
headphones
0
2000
4000
6000
8000
10000
12000
count

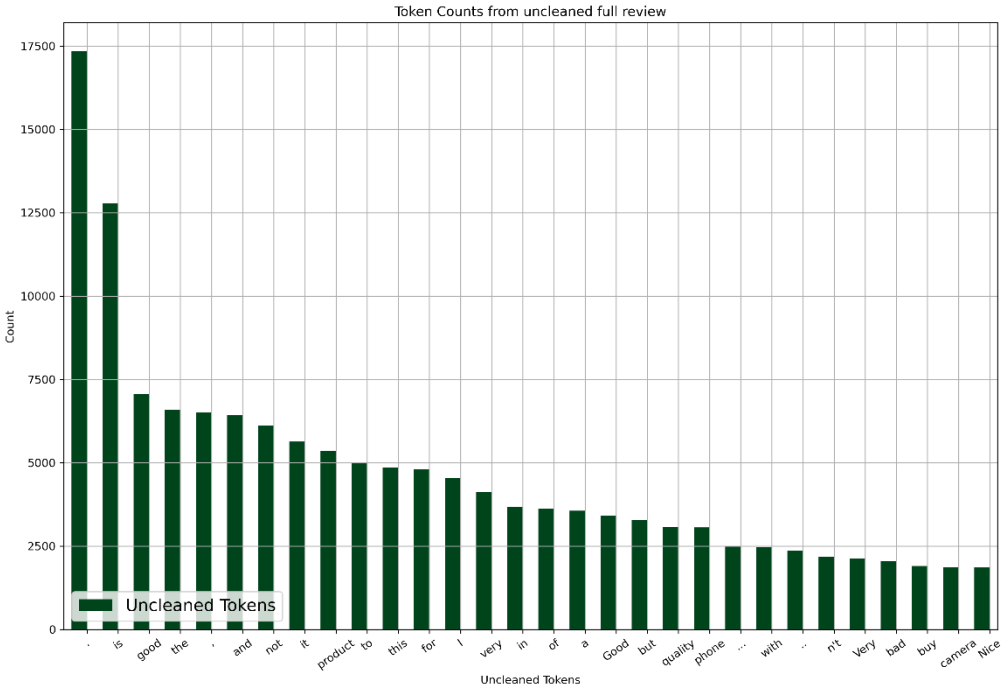

4000
3500
3000
2500
2000
1500
1000
500
0
Count
0
100
200
300
400
cleaned_review_length


Count
3000
2500
2000
1500
1000
500
0
0
100
200
300
400
500
length_of_uncleaned_review


gadget
phones
smart watch
printer
router
headphones
0
500
1000
1500
2000
2500
3000
count
Rating
0
1
2
3
4


Rating
0
1
2
3
4
full_review_word_range
10.0
30.0
100.0
200.0
0
500
1000
1500
2000
count


Token Counts from cleaned full review
Count
Cleaned Tokens
Cleaned Tokens
good
product
phone
quality
camera
nice
bad
buy
work
battery
use
watch
money
price
get
one
mobile
flipkart
range
best
poor
router
ok
display
time
worst
also
like
performance
sound



Observations:

1. Dataset contains 4 null values.
2. Word length of Summary is minimum
3. After removing unnecessary columns and null values the shape of the dataset is 28001 rows and 4 columns.
4. The dataset is multi-class type. It contains 5 different class which indicate rating of the full review.
5. Summary has 1204 unique type and the data scrapped is of 5 different gadgets categories reviews.
6. Only Rating column is in numerical form rest all columns are in text format.
7. Average length of uncleaned review is 68.6 words where as average length of cleaned review is 44.6 words.
8. Majority of the data scrapped is of phone electronic followed by watches and printers respectively. Router data is minor in the dataset.
9. Router has the minimum one-star rating where as phones have maximum five-star ratings. Printers have the least five -star rating.
10. Review with more word count has less rating comparatively.
11. Good and product are most used words in the review.
12. Product, bad and poor are the most used words for one and two-star rating.
13. Good is almost common and top most used word in all the ratings.
14. Nice, quality, awesome, product some of the top words in 4 and 5-star ratings.
15. Money, waste and worthless are top most used in 1-star summary.
16. Awesome, terrific are top words used in 5-star summary.

Results:

In the whole dataset the top words which were visualized must have the most influence on the label columns.

Conclusion:

Linear SVC machine learning model has performed better than all the other model as it has better accuracy and precision score for every label compared to other machine learning models. Also, True Positive and True Negative values for every label is in better balance for Linear SVC model.

Outcome of the Study:

Visualizing data helped to explore and study the data. Data Cleaning helps in minimizing the overfitting created during model training and improves the model performance. Random Forest can neglect outliers even when the data is fed with outliers to the machine learning model but it is not worth the use in this type of problem as it required ample amount of time and has underperformed. Simplifying the data was the most challenge part in this project but it can be overcome if minimal domain knowledge in NLP is exploited into this project.

Limitations of this work and Scope for Future Work:

The label predicted are only limited to particular electronic gadgets. When we try to predict for other type electronic gadgets the machine learning model will fail. To improve the model efficiency, we can add more of type of gadgets and increase the size of the data in order to balance the data and get much improved precision, recall and f1-score for the given model. Also, by removing more unnecessary words from the data which are not affecting the output data the model can be improvised.