Department of Computer Engineering

Vivekanand Education Society's Institute of Technology

2020-21

BIG DATA AND ANALYTICS

Mini project Report on

"Analysis of Amazon Product Review"

By

Anish Adnani (D17B - 01)

Nihal Bhandary (D17B - 09)

Supriya Patil (D17B - 50)

Under the Guidance of

Mrs Sujata Khedkar

Name of Lab Teacher

Mrs. Vidya Pujari

1>PROBLEM DEFINITION AND SCOPE OF THE PROJECT

1.1> INTRODUCTION

An existential problem for most of the online e-commerce website is to recommend the best quality products to its customers. Amazon, which is one of the largest e-commerce platforms, allows its users to add reviews to products that they bought from their portal. Based upon the reviews in our dataset we can easily identify which are most commonly used words in positive and negative reviews.

1.2> PROBLEM DEFINITION AND SCOPE OF PROJECT

In this project we aim to find out the most commonly used words in positive and negative reviews. Users sometimes by mistake have a good experience with the product and write a positive review and by mistake add less star rating. Identifying most frequently used words in the review will help us overcome this problem very easily. Even if there is an entry mistake by the user, we can easily classify the reviews as “positive” or “negative” based upon the tokens of the review and matching it with both “most frequently used words in positive review” and “most frequently used words in negative review”.

1.3> USERS OF THE SYSTEM

Users of the system involve

1. Seller selling his product on the portal
2. User who is purchasing the product and writing a review

1.4> DATASET

For our analysis we are using an open source dataset available on Kaggle.
Dataset link :
https://www.kaggle.com/eswarchandt/amazon-music-reviews?select=Musical_instruments_reviews.csv

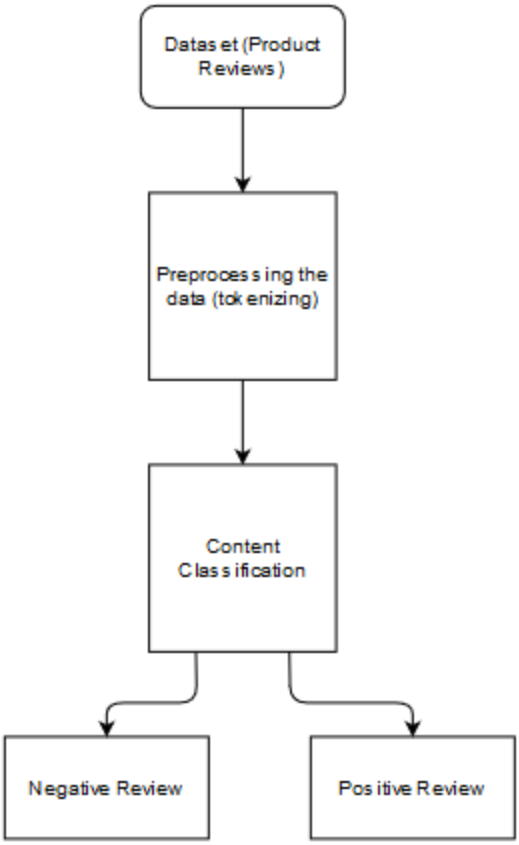
2> LITERATURE SURVEY

In 2018, Amazon’s share of the US ecommerce market hit 49%. That's 5% of all retail spent across the entire country. Almost 9 out of 10 prices check a product on Amazon. Amazon sells more than 12 million products a year and more than 95 million people have Amazon Prime membership. Amazon sells over 1.1 Million home improvement products. These staggering figures show how much data Amazon collects in a year and erroneous data is a dark truth of

any platform. Keeping the figures in mind we decided to overcome this problem by checking the review entered by the user as there is 19% chance that a human will make some error while entering the data online due to maybe lack of knowledge of technology, device issue, connectivity issues etc.

3> Conceptual System Design

3.1>Conceptual System Design- CSD Diagram with explanation of each module



Dataset : -
This module contains the reviews of the products by the user, this data is an unprocessed segment of our model forming the first step in the process.

Pre-Processing:-
Given a character sequence and a defined document unit (blurb of texts), tokenization is the task of chopping it up into pieces, called ***tokens***, perhaps at the same time throwing away certain characters/words, such as punctuation.

Content Classification:-
From the positive and negative reviews, most commonly used tokens are found. When a review is sent here for classification, depending on the tokens of that review it gets classified as positive or negative review.

3.2> Methodology

3.2.1 Data Gathering / Loading

We are going to classify Amazon product reviews to understand the positive or negative review. Amazon has a 5-star rating. We will use that to compare our prediction.

```
print(raw_data.shape)
```

```
(10261, 11)
```

```
raw_data.head()
```

	reviewerID	asin	reviewerName	helpful	reviewText	overall	summary	unixReviewTime	reviewTime	Unnamed: 9	target
0	A2IBPI20UZIR0U	1384719342	cassandra tu "Yeah, well, that's just like, u...	[0, 0]	Not much to write about here, but it does exac...	5	good	1393545600	02 28, 2014	NaN	1
1	A14VAT5EAX3D9S	1384719342	Jake	[13, 14]	The product does exactly as it should and is q...	5	Jake	1363392000	03 16, 2013	NaN	1
2	A195EZSQDW3E21	1384719342	Rick Bennette "Rick Bennette"	[1, 1]	The primary job of this device is to block the...	5	It Does The Job Well	1377648000	08 28, 2013	NaN	1
3	A2C00NNG1ZQQG2	1384719342	RustyBill "Sunday Rocker"	[0, 0]	Nice windscreen protects my MXL mic and preven...	5	GOOD WINDSCREEN FOR THE MONEY	1392336000	02 14, 2014	NaN	1
4	A94QU4C90B1AX	1384719342	SEAN MASLANKA	[0, 0]	This pop filter is great. It looks and perform...	5	No more pops when I record my vocals.	1392940800	02 21, 2014	NaN	1

3.2.2 Data Preprocessing ,Descriptive Analysis

A product review is a bunch of sentences posted by the user. Before the classification should take place, the sentence is broken down into meaningful units. Which then can be used to classify the review as positive or negative.
Text segmentation is the process of dividing written text into meaningful units, such as words, sentences, or topics. The term applies both to mental processes used by humans when reading text, and to artificial processes implemented in computers, which are the subject of natural language processing.

Tokenization is the process of demarcating and possibly classifying sections of a string of input characters. The resulting tokens are then passed on to some other form of processing. The process can be considered a sub-task of parsing input.

For example, in the text string:

The quick brown fox jumps over the lazy dog
the string isn't implicitly segmented on spaces, as a natural language speaker would do. The raw input, the 43 characters, must be explicitly split into the 9 tokens with a given space delimiter

3.2.3 Classification

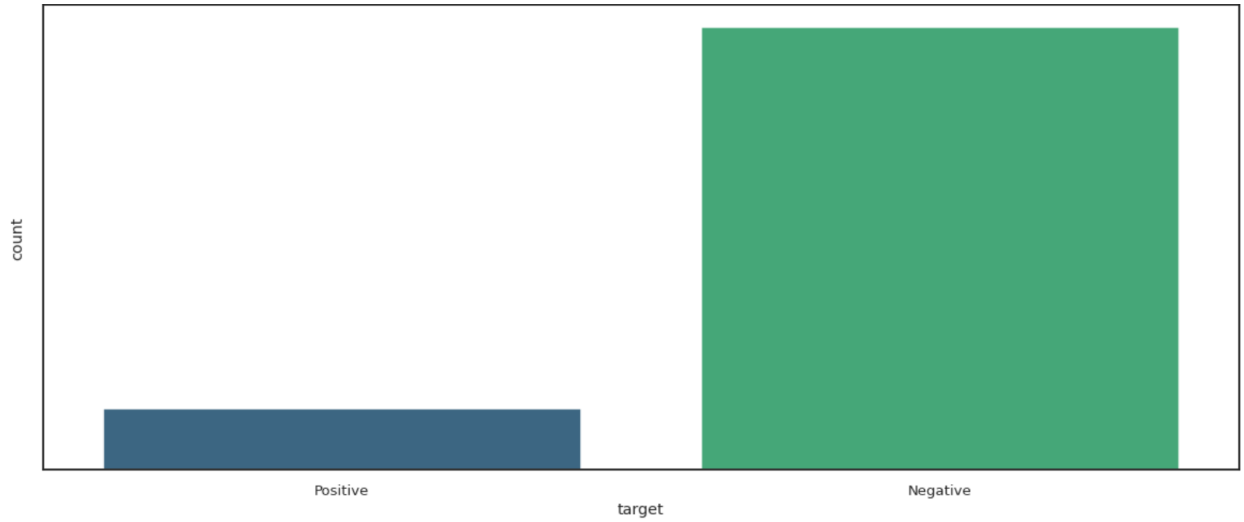
Once the tokens from positive and negative reviews are identified and stored the classification takes place. Input review is broken down into the tokens. The tokens are then identified and searched against the negative and positive list of tokens,

The most number of tokens matched alongside with the error rate, gives us the final output from the system.

3.2.4 Visualizations

Visualizing the Data

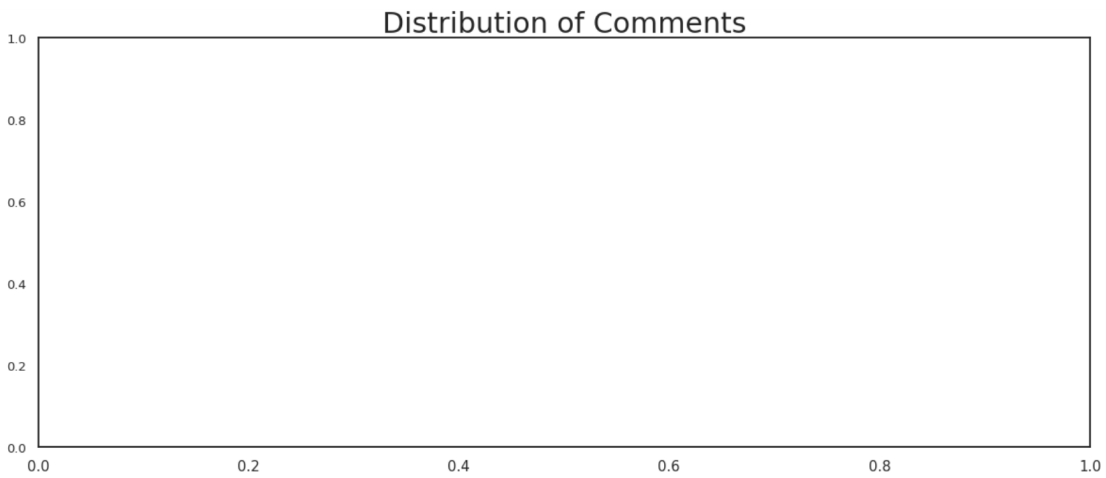
```
fig, ax = plt.subplots()
g = sns.countplot(raw_data.target, palette='viridis')
g.set_xticklabels(['Positive', 'Negative'])
g.set_yticklabels([])
```




```
# function to show values on bars
def show_values_on_bars(axs):
  def _show_on_single_plot(ax):
    for p in ax.patches:
      _x = p.get_x() + p.get_width() / 2
      _y = p.get_y() + p.get_height()
      value = '{:.0f}'.format(p.get_height())
      ax.text(_x, _y, value, ha="center")

  if isinstance(axs, np.ndarray):
    for idx, ax in np.ndenumerate(axs):
      _show_on_single_plot(ax)
  else:
    _show_on_single_plot(axs)

show_values_on_bars(ax)
sns.despine(left=True, bottom=True)
plt.xlabel('')
plt.ylabel('')
plt.title('Distribution of Comments', fontsize=30)
plt.tick_params(axis='x', which='major', labelsize=15)
fig.savefig('classes.png')
plt.show()
```




```
(len(raw_data.loc[raw_data.target==1])) / (len(raw_data.loc[raw_data.target == 0])) * 100
```

```
728.1678773204197
```

```
import random
```

```
index = random.sample(raw_data.index[raw_data.target == 1].tolist(), 5)
for i in index:
  print(raw_data.iloc[i, 6])
```

```
Love it
Give em some time, you'll like em!
Truly one of the best.
Everything is in order now
Nice value!
```

```
df = raw_data.sample(frac=0.3)
df.shape
```

```
(3078, 11)
```

4> TECHNOLOGY USED

We have used Google Colab for our implementation. Apart from this we are also using few libraries

1.numpy
2.pandas
3.seaborn
4.matplotlib
5.nltk
6.string

5> IMPLEMENTATION:

Implementing various packages and libraries

```
import numpy as np
print("ok")
```

```
ok
```

```
import pandas as pd
print("ok")
```

```
ok
```

```
import matplotlib.pyplot as plt
print("ok")
```

```
ok
```

```
import seaborn as sns
print("ok")
```

```
ok
```

```
[ ]  import nltk
     print("ok")
```

```
     ok
```

```
import re
print("ok")
```

```
ok
```

```
import string
```

```
from nltk.corpus import stopwords
```

```
from collections import Counter
print("ok")
```

```
ok
```

+

```
np.random.seed(27)
```

Setting matplotlib parameters

```
%matplotlib inline
```

```
plt.rcParams['figure.figsize'] = [20.0, 8.0]
```

```
sns.set_palette('viridis')
sns.set_style('white')
sns.set_context('talk', font_scale=0.8)
print("ok")
```

```
ok
```

Uploading the Data

```
from google.colab import files
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving Musical_instruments_reviews.csv to Musical_instruments_reviews.csv
```

```
import io
```

```
raw_data = pd.read_csv(io.BytesIO(uploaded['Musical_instruments_reviews.csv']))
print("ok")
```

```
ok
```

Separating the tokens

```
nlp = spacy.load('en')
```

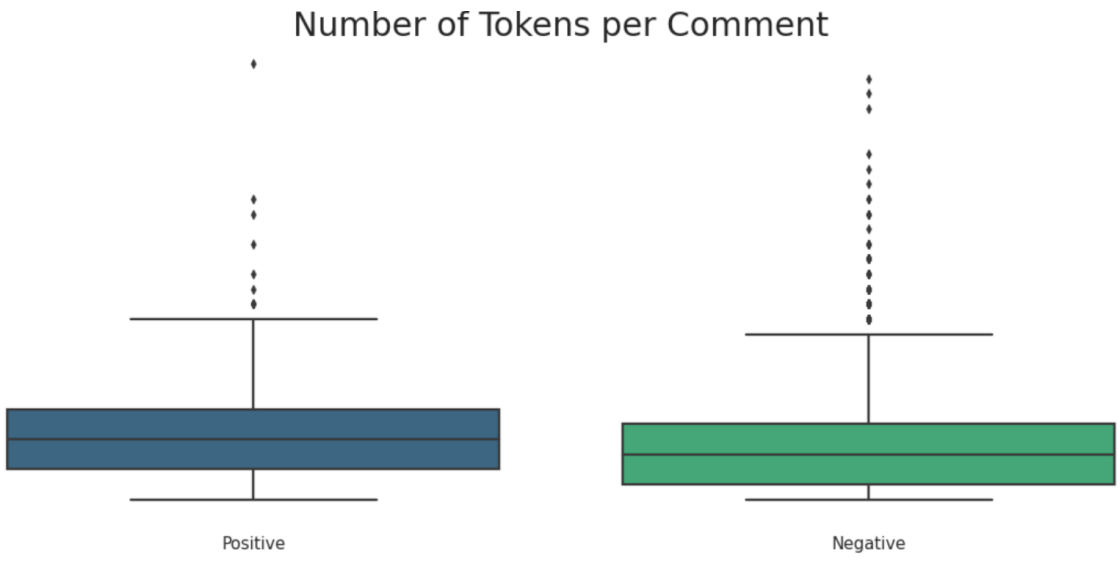
```
df['tokens'] = [nlp(text, # disable parts of the language processing
#pipeline we don't need here to speed up processing
 disable=['ner', # named entity recognition
 'tagger', # part-of-speech tagger
 'textcat', # document label categorizer
 ]) for text in df.summary]
df.sample(5)
print("ok")
```

ok

```
df['num_tokens'] = [len(token) for token in df.tokens]
df.sample(5)
```

	reviewerID	asin	reviewerName	helpful	reviewText	overall	summary	unixReviewTime	reviewTime	Unnamed: 9	target	tokens	num_tokens
0	A3CUYX2VXZM9KO	B000J5XS3C	Jay	[0, 0]	the Beringer 802 mixer is by far the lowest co...	5	A Lot of Bang for the Buck !!	1355443200	12 14, 2012	NaN	1	(A, Lot, of, Bang, for, the, Buck, !, !)	9
1	A8MQNZK3J0UEX	B000VJ2VCK	Paul C.	[0, 0]	Pretty straight forward bench. I was surprised...	5	Cool product	1329868800	02 22, 2012	NaN	1	(Cool, product)	2
2	A3MRBUCGH66JTY	B0002F7IN0	K. Andrews "Permanent tourist"	[1, 1]	I'm very happy with this stand. It is heavy, ...	5	Sturdy and well made	1332720000	03 26, 2012	NaN	1	(Sturdy, and, well, made)	4
3	A2Y4PRG2HHOL3G	B0009G1E0K	'brother' Bill	[0, 0]	"Nice for the price" ... practical, ev...	3	Cheap & Good	1387584000	12 21, 2013	NaN	0	(Cheap, &, Good)	3
4	A10FM4ILBIMJJ7	B005PGGU9O	Sim	[0, 0]	This appears to be a good sturdy capo with a s...	5	Great for the money spent.	1381795200	10 15, 2013	NaN	1	(Great, for, the, money, spent, .)	6

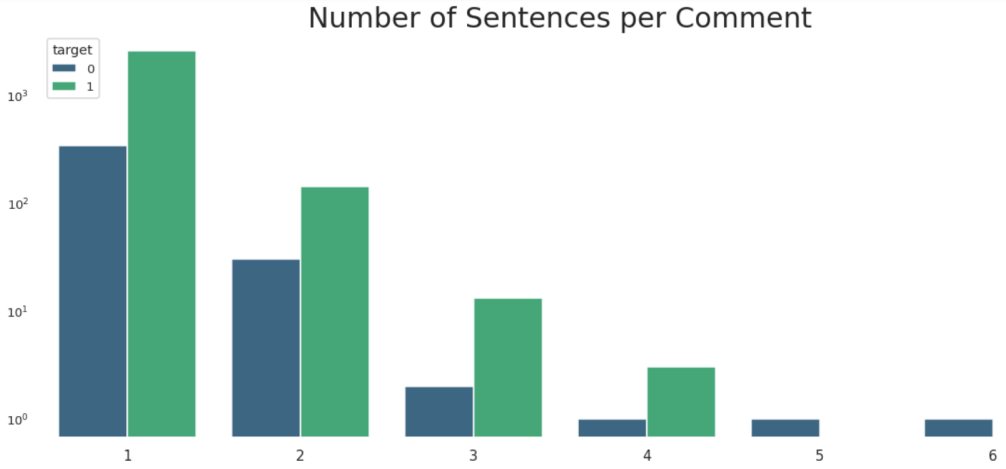
```
fig, ax = plt.subplots()
g = sns.boxplot(x=df.target, y=df.num_tokens, palette='viridis')
g.set_xticklabels(['Positive', 'Negative'])
g.set_yticklabels([])
sns.despine(left=True, bottom=True)
plt.xlabel('')
plt.ylabel('')
plt.title('Number of Tokens per Comment', fontsize=30)
plt.tick_params(axis='x', which='major', labelsize=15)
fig.savefig('tokens.png')
plt.show()
```




```
sents = [list(x.sents) for x in df.tokens]
df['num_sents'] = [len(sent) for sent in sents]
df.sample(5)
```

	reviewerID	asin	reviewerName	helpful	reviewText	overall	summary	unixReviewTime	reviewTime	Unnamed: 9	target	tokens	num
9608	AAHV0LYDMZGY8	B0078YF9V6	Bryan Harvey	[10, 12]	Okay... just strung these up and ran them thro...	3	a little too bright and high output for classi...	1332028800	03 18, 2012	NaN	0	(a, little, too, bright, and, high, output, fo...	
3484	A28DVY0Q6PPO1J	B0002PBS6S	Tatiana Montenegro	[0, 0]	the only strings I get are Ernie Ball strings....	5	strings	1360540800	02 11, 2013	NaN	1	(strings)	
6799	A2MPM6M93OXIJT	B001PGXHX0	Linda Noullet	[0, 0]	I have a few different brands of picks at home...	5	A nice pick - good feel	1330128000	02 25, 2012	NaN	1	(A, nice, pick, -, good, feel)	
7993	A18C03TA8N26L4	B003K16N0C	Javier Fernandez	[2, 2]	Nice presentation, although the harmonicas are...	4	Lots of fun	1309910400	07 6, 2011	NaN	1	(Lots, of, fun)	
51	A2U1Z3TZ4P76JB	B000068NW5	C. Zemer	[0, 0]	Bought this for my daughter along with her new...	5	Cable works as described	1393977600	03 5, 2014	NaN	1	(Cable, works, as, described)	

```
fig, ax = plt.subplots()
g = sns.countplot(df.num_sents, hue=df.target, palette='viridis')
#g.set_xticklabels(['Positive', 'Negative'])
g.set_yticklabels([])
# using log scale on y-axis so we can better see the questions with more
#sentences
ax.set(yscale='log')
sns.despine(left=True, bottom=True)
plt.xlabel('')
plt.ylabel('')
plt.title('Number of Sentences per Comment', fontsize=30)
plt.tick_params(axis='x', which='major', labelsize=15)
fig.savefig('comment.png')
plt.show()
```




```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

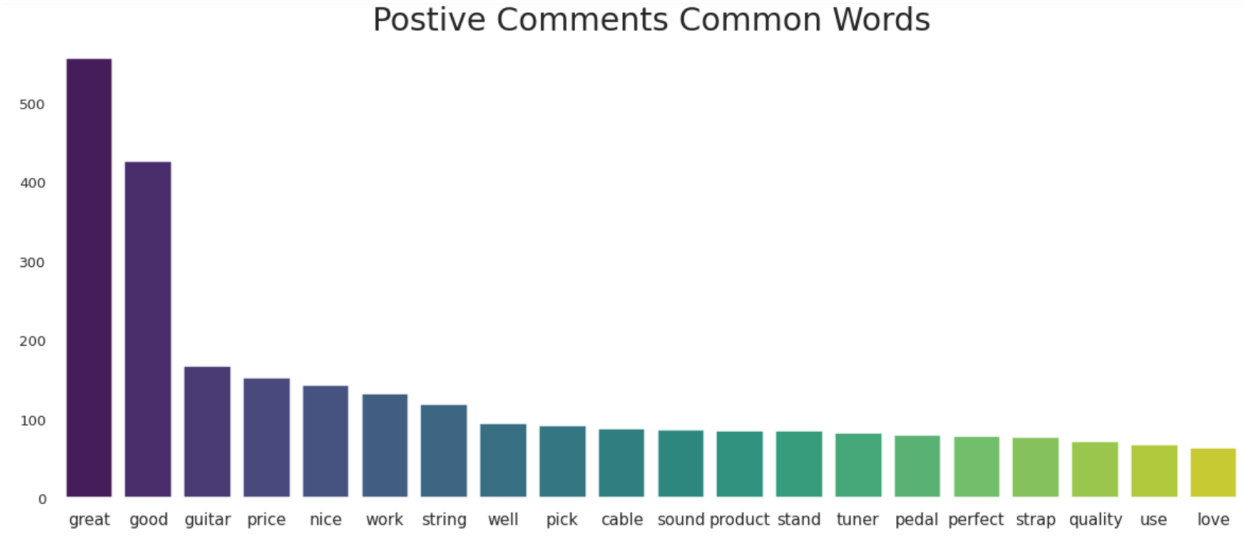
```
punctuations = string.punctuation
stop_words = set(stopwords.words("english"))
```

Calculating the most frequently used tokens in positive and negative reviews

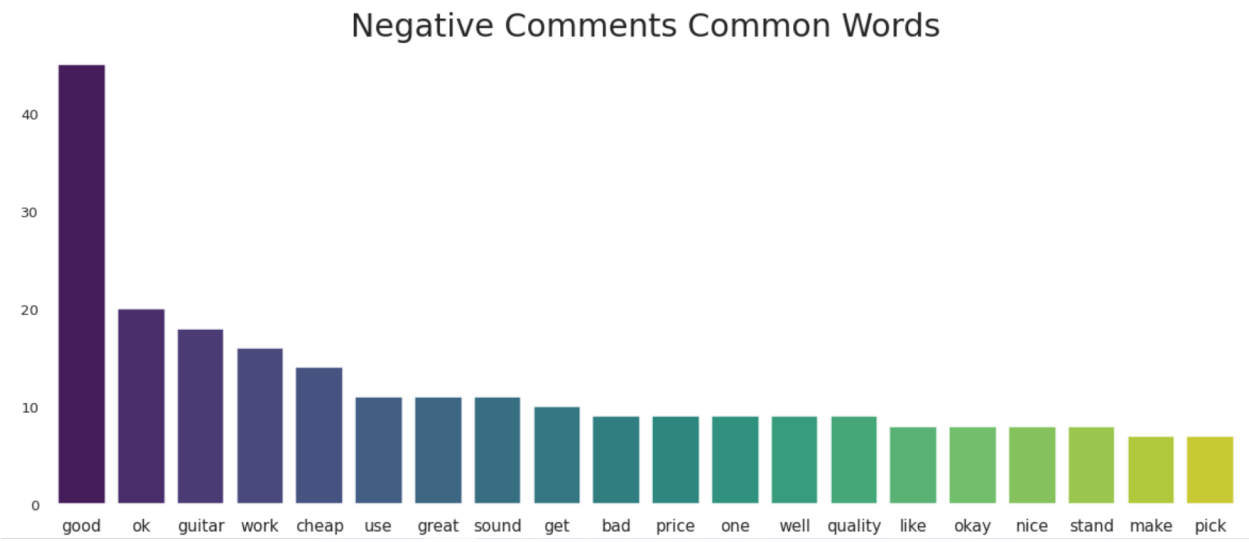
```
def cleanup_text(docs):
  texts = []
  for doc in docs:
    doc = re.sub(r'[^a-zA-Z\s]', '', doc, re.I|re.A)
    doc = nlp(doc, disable=['ner'])
    tokens = [tok.lemma_.lower().strip() for tok in doc if tok.lemma_!= '-PRON-']
    tokens = [tok for tok in tokens if tok not in stop_words and tok not in punctuations]
    tokens = ' '.join(tokens)
    texts.append(tokens)
  return pd.Series(texts)
print("ok")
positive_text = [text for text in df[df['target'] == 1]['summary']]
positive_clean = cleanup_text(positive_text)
positive_clean = ' '.join(positive_clean).split()
positive_counts = Counter(positive_clean)
positive_common_words = [word[0] for word in
positive_counts.most_common(20)]
positive_common_counts = [word[1] for word in
positive_counts.most_common(20)]
sns.barplot(positive_common_words, positive_common_counts,
palette='viridis')
sns.despine(left=True, bottom=True)
plt.xlabel('')
plt.ylabel('')
plt.title('Postive Comments Common Words', fontsize=30)
plt.tick_params(axis='x', which='major', labelsize=15)
fig.savefig('positive_words.png')
```

```
plt.show()
```

```
ok
/usr/local/lib/python3.6/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12,
  FutureWarning
```




```
negative_text = [text for text in df[df['target'] == 0]['summary']]
negative_clean = cleanup_text(negative_text)
negative_clean = ' '.join(negative_clean).split()
negative_counts = Counter(negative_clean)
negative_common_words = [word[0] for word in
negative_counts.most_common(20)]
negative_common_counts = [word[1] for word in
negative_counts.most_common(20)]
sns.barplot(negative_common_words, negative_common_counts,
palette='viridis')
sns.despine(left=True, bottom=True)
plt.xlabel('')
plt.ylabel('')
plt.title('Negative Comments Common Words', fontsize=30)
plt.tick_params(axis='x', which='major', labelsize=15)
fig.savefig('negative_words.png')
plt.show()
```



6> Result and Conclusion

Thus by using various nlp libraries we have successfully found out most frequently used words in positive and negative reviews. This could be used to solve the problem of counterfeit reviews or error while reporting a review. Implementation of such a procedure would result in much better product recommendations to its users.