Amazon Review Analysis

A Project Documentation

Submitted by

Sai Kiran

Vivek Kadam

Yogesh Pandiya

Yogesh Yadav

In partial fulfilment of the requirements for certification in

Advanced PGP in Data Science and Machine Learning

Data Science and Machine Learning

Jan 2023

ACKNOWLEDGEMENT

We have deep sense of gratitude and profound thanks to our project supervisor Dr. Amit Kumar and all mentors who have constantly guided, encouraged and motivated us in this journey. They made the journey more interesting and ensured we learn concepts really well.

We would like to thank Mr. Kshitij Upadhyay for laying the strongest foundation in the field of data science anyone could wish for and going the extra mile to answer our questions comprehensively.

We would also like to show our gratitude towards NIIT for designing this wonderful course and giving us the chance to work on this project

Sai Kiran

Vivek Kadam

Yogesh Pandiya

Yogesh Yadav

INDEX

1. INTRODUCTION

The daily changing technology have revolutionised the way we do shopping. When buying products online we cannot physically experience the product and daily different brands are launching new products. Therefore, the customers rely on the reviews written by other user to know the product performance while considering the product. The customer cannot go through the entire text reviews so he can rely on the average user rating and see how many verified buyers reviewed that to build his confidence in the product and that's why the reviews become so important for any e-commerce platform and they need to provide best rated product to improve the customers experience on platform and retain them.

1.1 Objective:

During their decision-making process, consumers want to find useful reviews as quickly as possible by rating system. Therefore, models able to predict the user rating from the text review are critically important. Getting an overall sense of a textual review could in turn improve consumer experience.
To develop an automated system to analyse and monitor an enormous number of reviews. Analyse tone, language, keywords and trend to provide valuable insight that increases the success rate of marketing campaigns of new and existing products.

After treating the data, doing data processing and applying various ML algorithms, time-series NLP models to get the insight from the data to help company grow the business.

The main goal for this project is to develop a model to predict user rating, usefulness of review and recommend most similar items to users based on collaborative filtering.

2. DATA COLLECTION

For this project, the Industrial and Scientific dataset consist of reviews and product information from amazon were collected. This dataset includes reviews (ratings, text, summary, productID) and product metadata (descriptions, category information, price and brand).

2.1 Product Complete Reviews data:

This data contains the review about the product such as (product ID, review, summary, rating),
The link for dataset is: https://nijianmo.github.io/amazon/index.html
The original data was in json format, we read the data using a method of pandas (pd.read_json()) and converted the data into csv format after cleaning the data.
The sample of this data looks like as showcased below:

	overall	verified	reviewTime	reviewerID	asin	style	reviewerName	reviewText	summary	unixReviewTime	vote	image
0	5	True	11 27, 2017	A1JB7HFWHRYHT7	B0000223SI	{'Size:': ' 1-(Pack)'}	Alex W.	This worked really well for what I used it for...	Couldn't have been happier with it's performance	1511740800	NaN	NaN
1	5	True	11 4, 2017	A2FCLJG5GV8SD6	B0000223SI	{'Size:': ' 1-(Pack)'}	Randall Harris	Fast cutting and good adheasive.	Good paper.	1509753600	NaN	NaN
2	5	False	10 27, 2017	A3IT9B33NWYQSL	B0000223SI	{'Size:': ' 1-(Pack)'}	A. C.	Worked great for my lapping bench. I would li...	Handy!	1509062400	NaN	NaN
3	4	True	01 13, 2018	AUL5LCV4TT73P	B0000223SK	{'Size:': ' 1-Pack'}	TnT	As advertised	As advertised	1515801600	NaN	NaN
4	5	True	10 7, 2017	A1V3I3L5JKO7TM	B0000223SK	{'Size:': ' 1-Pack'}	John Jones	seems like a pretty good value as opposed to b...	seems like a pretty good value as opposed to b...	1507334400	NaN	NaN

Each row corresponds to a customer review and include the following variables :

ReviewerID: ID of the reviewer, e.g. A1JB7HFWHRYHT7- type: object

asin : ID of the product , e.g., B0000223SI– type: object

reviewerName : name of the reviewer – type: object

reviewText : text of the review – type: object

overall : Rating (1,2,3,4,5)– type: float64

summary : summary of the review – type: object

unixReviewTime : time of the review (unix time) –type:int64

reviewTime : time of the review (raw) – type: object

2.2 **Product Metadata:**

This dataset includes electronics product metadata such as descriptions, category information, price, brand, and image features. This dataset was obtained from http://jmcauley.ucsd.edu/data/amazon/. The json was imported and decoded to convert json format to csv format after merging and cleaning The sample product meta dataset is shown below:

category	tech1	description	fit	title	also_buy	tech2	brand	feature	rank	also_view	main_cat	similar_item	date	price	asin
[Industrial & Scientific, Industrial Electrica...		[RCRF-03 - Works on all Turning Technologies R...		Turning Technologies Response Card (RCRF-03)	[1454894547, 0133455548, 0538741252, 193493139...		Turning Technologies, LLC	[]	12,329 in Industrial & Scientific (	[]	Industrial & Scientific			$23.61	0176496920
[]		[The only laminated pocket tool that provides ...		R-Cat 692782109 EKG Badge with Arrhythmia Pock...	[B01NBCNTJ9, B01IAHKULS, B01MT59BRR, 194100406...		R-Cat	[]	20,174 in Industrial & Scientific (	[]	Industrial & Scientific			$20.00	0692782109
[Industrial & Scientific, Science Education]	class="a-keyvalue prodDetTable" role="present...	[, Now in its Second Edition, <B>The Spina...		Anatomical Chart Company's Illustrated Pocket ...	[0781776821, 078177683X, 0781776783, 157222684...		Anatomical Chart Company	[MPN: 9780781776844, Authentic Anatomical Char...	[>#351,522 in Office Products (See top 100), >...	[]	Office Products		August 7, 2007	$10.37	0781776848
[Industrial & Scientific, Science Education, C...	class="a-keyvalue prodDetTable" role="present...	[, Developed in conjunction with a health...		Joints of the Lower Extremities Anatomical Chart	[0781786630, 1587798085, B004MAJHLW, 078178656...		Anatomical Chart Company	[MPN: 9780781786607, Authentic Anatomical Char...	[>#459,493 in Office Products (See top 100), >...	[B07BCNHC3K, B07D6X5TCB, 1587799839]	Office Products		February 26, 2009		0781786606

Each row corresponds to product and includes the variables:

asin: ID of the product, e.g. B0000223SI

title: name of the product

price: price in US dollars (at time of crawl)

imUrl: url of the product image

related: related products (also bought, also viewed)

salesRank: sales rank information

brand: brand name

categories: list of categories the product belongs to

3. DATA WRANGLING

Data wrangling, also known as data munging or data preparation, is the process of transforming raw data into a format that is suitable for analysis and modelling. It involves a variety of tasks such as cleaning, transforming, and enriching the data. The goal of data wrangling is to make sure that the data is in a format that is consistent, accurate, and complete, and that it can be easily integrated with other data sources.

1. Data cleaning: This involves removing or correcting errors, inconsistencies, and missing values in the data.
2. Data transformation: This involves changing the format, structure, and representation of the data to make it more usable. This can include tasks such as normalizing data, encoding categorical variables, and creating new features.
3. Data integration: This involves combining data from multiple sources into a single dataset. This can include tasks such as joining tables, merging datasets, and aggregating data.

3.1 Merging DataFrames ;

These two files were imported using pandas read_json into dataframes to work on and these two dataframes were merged using the inner join on common column which is 'asin'(product ID) and the dataframe was named as data_df. The sample of data_df is showcased below:

verified	reviewTime	reviewerID	asin	style	reviewerName	reviewText	summary	unixReviewTime	vote	image	category	tech1	description	fit
True	11 27, 2017	A1JB7HFWHRYHT7	B0000223SI	{'Size:': ' 1-(Pack)'}	Alex W.	This worked really well for what I used it for...	Couldn't have been happier with it's performance	1511740800	NaN	NaN	[Industrial & Scientific, Abrasive & Finishing...	class="a-keyvalue prodDetTable" role="present...	[Amazon.com This superior quality, clog- and l...	
True	11 27, 2017	A1JB7HFWHRYHT7	B0000223SI	{'Size:': ' 1-(Pack)'}	Alex W.	This worked really well for what I used it for...	Couldn't have been happier with it's performance	1511740800	NaN	NaN	[Industrial & Scientific, Abrasive & Finishing...	class="a-keyvalue prodDetTable" role="present...	[Amazon.com This superior quality, clog- and l...	
True	11 4, 2017	A2FCLJG5GV8SD6	B0000223SI	{'Size:': ' 1-(Pack)'}	Randall Harris	Fast cutting and good adheasive.	Good paper.	1509753600	NaN	NaN	[Industrial & Scientific, Abrasive & Finishing...	class="a-keyvalue prodDetTable" role="present...	[Amazon.com This superior quality, clog- and l...	
True	11 4, 2017	A2FCLJG5GV8SD6	B0000223SI	{'Size:': ' 1-(Pack)'}	Randall Harris	Fast cutting and good adheasive.	Good paper.	1509753600	NaN	NaN	[Industrial & Scientific, Abrasive & Finishing...	class="a-keyvalue prodDetTable" role="present...	[Amazon.com This superior quality, clog- and l...	

Below showcased is the column description of the final data :

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 81520 entries, 0 to 81519
Data columns (total 30 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   overall          81520 non-null  int64 
 1   verified         81520 non-null  bool  
 2   reviewTime       81520 non-null  object
 3   reviewerID       81520 non-null  object
 4   asin             81520 non-null  object
 5   style            38339 non-null  object
 6   reviewerName     81490 non-null  object
 7   reviewText       81508 non-null  object
 8   summary          81509 non-null  object
 9   unixReviewTime   81520 non-null  int64 
 10  vote             10127 non-null  object
 11  image            1749 non-null   object
 12  category         81520 non-null  object
 13  tech1            81520 non-null  object
 14  description      81520 non-null  object
 15  fit              81520 non-null  object
 16  title            81520 non-null  object
 17  also_buy         81520 non-null  object
 18  tech2            81520 non-null  object
 19  brand            81520 non-null  object
 20  feature          81520 non-null  object
 21  rank             81520 non-null  object
 22  also_view        81520 non-null  object
 23  main_cat         81520 non-null  object
 24  similar_item     81520 non-null  object
 25  date             81520 non-null  object
 26  price            81520 non-null  object
 27  imageURL         81520 non-null  object
 28  imageURLHighRes  81520 non-null  object
 29  details          81520 non-null  object
dtypes: bool(1), int64(2), object(27)
memory usage: 18.7+ MB
```

After merging the data, we had **81520** rows and **30** columns

The data-types of few columns need to be changed so we can get meaning insights from the columns such as price, date and rank.

There are lot of missing values and empty list, string present in the data , the price column have paragraphs inside them and rank is given in between text so we need to process data to get meaningful insights from this raw data.

The Statistical description of raw data:

	overall	unixReviewTime
count	81520.000000	8.152000e+04
mean	4.526816	1.454085e+09
std	0.946186	4.627941e+07
min	1.000000	1.051402e+09
25%	4.000000	1.426982e+09
50%	5.000000	1.458778e+09
75%	5.000000	1.486512e+09
max	5.000000	1.538093e+09

3.2 Data Pre-processing:

a. Replacing empty string with NaN Values

Replacing the empty string values to NaN values so we do null value treatment and these values don't impact our ML models performance.

b. Null Value Column Treatment :

After doing the above process we had these many missing values in the data:-

```
overall                0
verified               0
reviewTime             0
reviewerID             0
asin                   0
style              43181
reviewerName          30
reviewText            12
summary               11
unixReviewTime         0
vote               71393
image              79771
category               0
tech1              49402
description            0
fit                81446
title                  0
also_buy               0
tech2              81448
brand                265
feature                0
rank                   0
also_view              0
main_cat              98
similar_item       75864
date               45087
price              11610
imageURL               0
imageURLHighRes        0
details                0
dtype: int64
```

To deal with so many null value in different columns we created a user defined function named as len_null to drop columns on the basis of number of values present in the columns and in some read data is in list and dictionary ,our loop reads inside the list and dictionary to check if they are empty and then count them as null values and return the unwanted columns who have more than 25% null values , as filling such columns is not advised as this will highly skew the data and hence we should drop them .

Before dropping these columns – we made two more user defined functions to tell the most frequent values present in the data and to see how many unique values and duplicate are there in these unwanted columns so we have confidence while dropping these columns.

Most of these columns had majority count as NaN values and duplicate value as **16901** in each columns so we got that idea that data have duplicates value and count of unique value is also less so we got the confidence in dropping these columns as we are not dropping anything significant and we can proceed ahead to drop these columns.

After dropping unwanted columns we had these columns with that many null values :

```
overall              0
verified             0
reviewerID           0
asin                 0
reviewerName        30
reviewText          12
summary             11
unixReviewTime       0
category             0
description          0
title                0
brand              265
rank              4731
main_cat            98
price            11610
dtype: int64
```

We didn't fill these null values here only as some of our columns have data in list that need to be taken into string form and columns such as price and rank need to be extracted into meaningful form and after doing these we will do missing value treatment.

c. **Finding Duplicate values :**

After reading above columns details we got the idea that our data have duplicate values .

The count of duplicate value is : **16901**.

We removed these value

d. Extracting data from list:

Some of our column have data inside list and we created a user defined function named as detailss to get the data in string and tells the count of unique values along with null values

e. Getting useful details from column of price and rank :

We used lambda function and regex methods (findall and sub) to get the meaning data from these two columns and changed the data type of these two columns from object to float.

After getting the data we have null value in these columns and we will impute later.

f. Extracting date in datatime format:

We used the pandas to_datetime method on `unixReviewTime` to get the date in date time format and use that for finding information patterns based on time.

The datatype of desired columns after processing are :-

```
overall                  int64
verified                  bool
reviewerID              object
asin                    object
reviewerName            object
reviewText              object
summary                 object
unixReviewTime    datetime64[ns]
category                object
description             object
title                   object
brand                   object
rank                   float64
main_cat                object
price                  float64
dtype: object
```

g. Missing value treatment :

Now once we have our data in meaning form we can do missing values treatment.

We filled the **11610** null values in price column by grouping the data on category and then filling the filling the missing value with the mean price of that category. We had **858** unique categories and we dropped some null values where count of missing values is less.

Data after treating missing values :-

```
overall           0
verified          0
reviewerID        0
asin              0
reviewerName      0
reviewText        0
summary           0
unixReviewTime    0
category          0
description       0
title             0
brand             0
rank              0
main_cat          0
price             0
dtype: int64
```

Our final data have **68561** rows and **15** columns to do analysis.

3.3 Descriptive Statistics :

The following summary statistics was obtainedNumber of reviews: 68561

Number of unique reviewers:11014

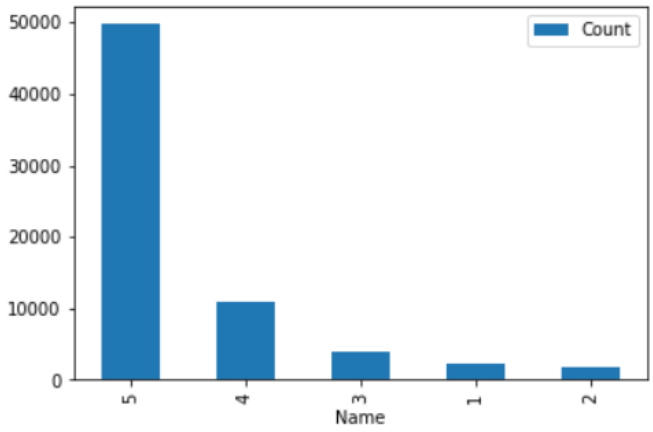
Number of unique products: 5105

Average rating score: 4.526816

	index	overall	verified	rank	price
count	68561.000000	68561.000000	68561.000000	68561.000000	68561.000000
mean	44704.511121	4.518925	0.919386	87197.255276	26.529923
std	22429.810431	0.953802	0.272244	145289.642202	85.688458
min	0.000000	1.000000	0.000000	100.000000	0.000000
25%	28042.000000	4.000000	1.000000	4223.000000	7.490000
50%	45785.000000	5.000000	1.000000	21800.000000	11.420000
75%	63986.000000	5.000000	1.000000	107482.000000	19.990000
max	81519.000000	5.000000	1.000000	993620.000000	2499.000000

Analysis of Ratings

```
The Number of unique values in column - overall is 5

The Null values in this column overall is 0

The number of duplicates are : 0

The Most frequent Value are :

The Unique Value are - [5 4 3 1 2]
```



3.4 ETL (Extract, Transform, Load)

It is a process used to move data from one or more sources into a target destination, such as a data warehouse or a data lake. It is a common approach used in data integration and data warehousing to manage and organize large volumes of data

- **Exporting the data into SQL database .**

- **Exporting the clean file to csv format so we don’t have to repeat all these steps every time.**

4. EXPLORATORY DATA ANALYSIS (EDA):

It is an approach to analysing data that is used to gain insights and understanding about the data. It is an iterative process that involves visualizing, summarizing, and discovering patterns in the data. EDA is typically the first step in the data analysis process and is used to gain a preliminary understanding of the data before building models or making predictions.

After Collecting data and wrangling data we did exploratory analysis of data , we did that analysis in 3 parts :

1. **Univariate data analysis** : exploring one column at time
2. **Bivariate data analysis** : exploring relationship between two columns
3. **Multivariate data analysis** : exploring pattern and relationship between multiple columns of the data .

We categorized data into categorical or continuous based on number of unique values in each column. We oriented number of unique values in each column and here is snippet of number of unique values in each column, where the number of unique values are less than 200 we considered them as categorical and our categorical columns are – overall , verified , category , main_category .Where as the price and rank were our continuous variable and rest of columns are either review and ID of product and reviewer.

Have a look at number of unique values in each column :

```
overall : 5
verified : 2
reviewerID : 11014
asin : 5105
reviewerName : 9761
reviewText : 55237
summary : 37486
unixReviewTime : 2894
category : 804
description : 4469
title : 5064
brand : 1842
rank : 4992
main_cat : 20
price : 1950
```

4.1 Insights:

- We also created a interact function where we can slide one by one and we can have a visual representation and information about that column.

- The interact function is a feature of the ipywidgets library in Python that allows you to create interactive widgets for controlling the behaviour of a function. It takes a function as an argument and creates widgets for each of the function's arguments, allowing you to change the arguments and see the results immediately

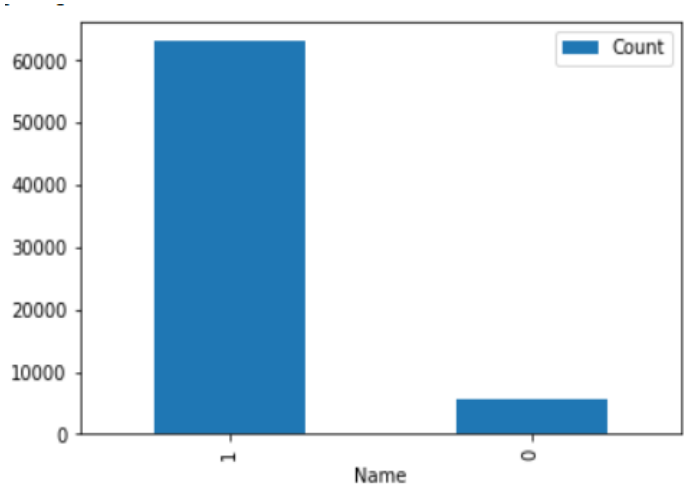
Overall : We see that our data have most reviews who rating is higher.

We can clearly see that people don't bother leaving a rating for a mediocre product, people who are extremely satisfied with their product (score of 5), have taken the time to leave a rating. Also it is seen that people have left more ratings for 1 star products than 2 star products, that shows people who are extremely dissatisfied with their purchase take more initiative than people who are just slightly unhappy with their purchase.

	Name	Count
0	5	49738
1	4	10884
2	3	3965
3	1	2247
4	2	1727

Verified : This column tells about whether the customer who is writing a review is verified or not

and we see that most of the reviews are written by those customers who have actually bought the product



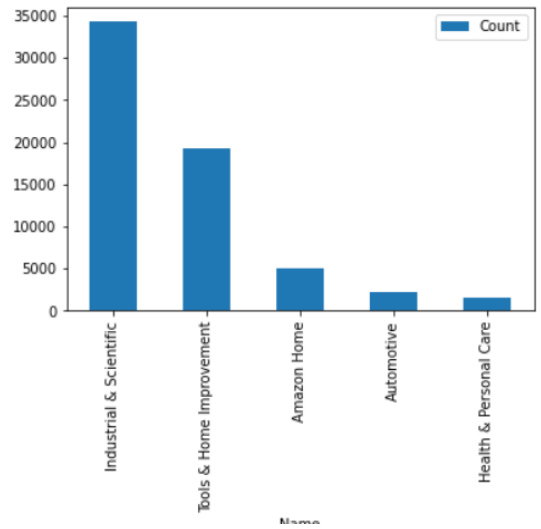
1 = Verified

0 = Not Verified

	Name	Count
0	1	63034
1	0	5527

Main_Category: This column tells about the category of the product the customer have written review for , There are **20** unique category in this column.

Our data have a mix of category but few **most frequent categories are –**



	Name	Count
0	Industrial & Scientific	34291
1	Tools & Home Improvement	19256
2	Amazon Home	5018
3	Automotive	2191
4	Health & Personal Care	1571

Category :

We have **804** unique categories and it's a mix of segments where the product can be described better.

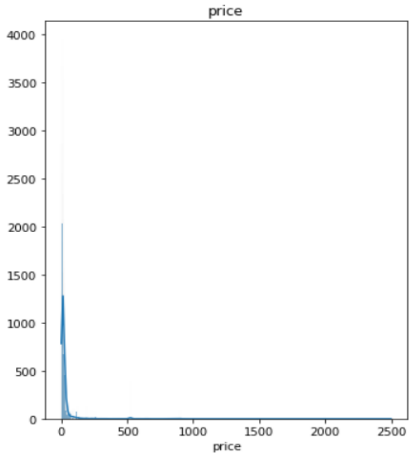
<u>Continuous variable</u> :

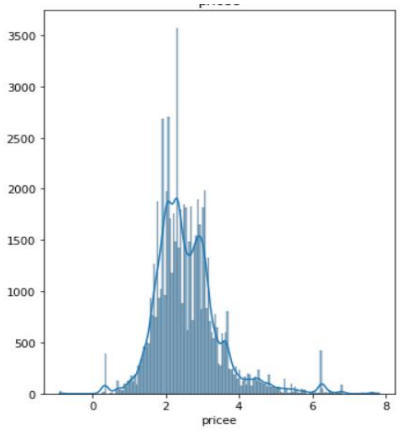
We also created a interact for the continuous variable which on slide gives you a histogram and box plot for the selected column along with some useful information.

<u>Price :</u>

This column tells about the price of product. Our price column has some outliers and to suppress the outlier we took log of the price and below showcased is the comparison of both.

```
The mean of the price is 26.529922696576772
The median of the price is 11.42
The standard deviation of the price is 85.68845816457652
```

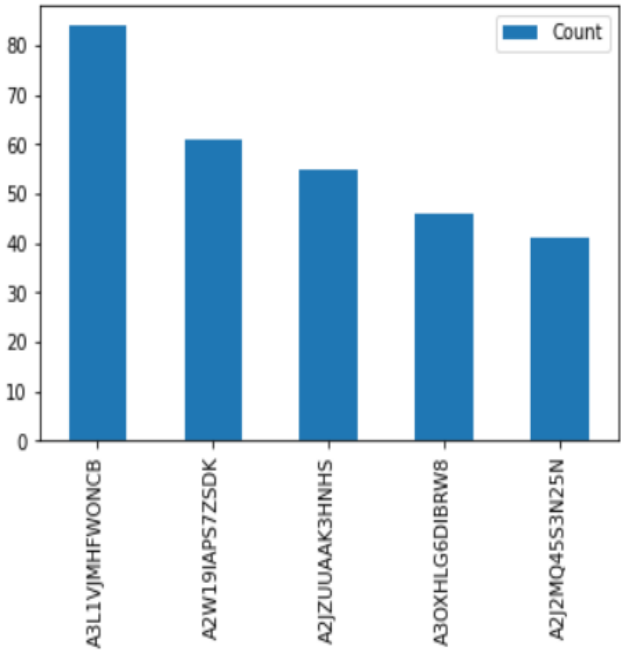




Reviewers:

There are **11014** unique customers who have written reviews for various product they bought from amazon.

Top 5 reviewerID :



	Name	Count
0	A3L1VJMHFWONCB	84
1	A2W19IAPS7ZSDK	61
2	A2JZUUAAK3HNHS	55
3	A3OXHLG6DIBRW8	46
4	A2J2MQ45S3N25N	41

Name of Top customers who buys item frequently and often writes reviews about that product are :

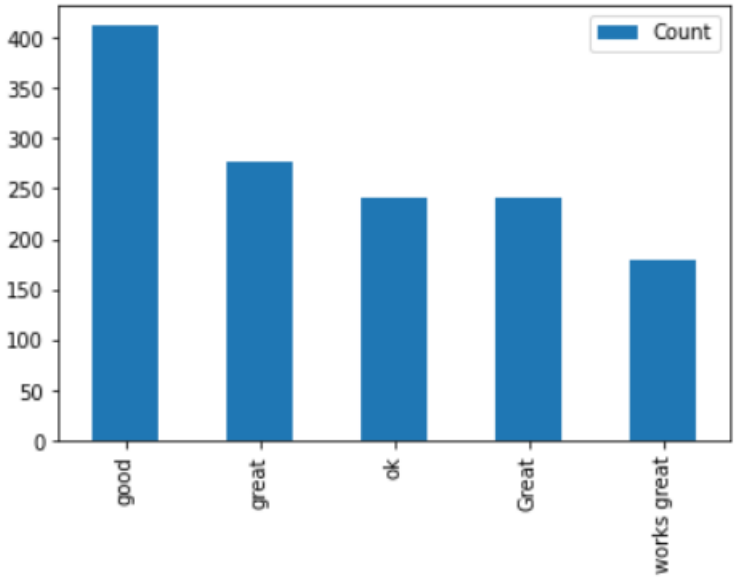
```
"['CDP&#039;s Onceagain habit of purchasing']"
"['Ian Mazursky']",
"['Old Sarge']",
"['C. Hill']",
"['Amazon Customer']",
"['Trex']",
"['ventingisok']",
"['sugarbear' 'pattiboypatrick16']",
"['SJK']",
"['Frog']"]
```

ReviewText :

This column have the review written by the customer who have bought items and there are also instances where the reviewer have written down review without buying the product and we will look into later .

The most frequent words in this column are good , great , works great which tells that the customers are liking the products and they have written positive reviews.

Most frequent words in this column are :

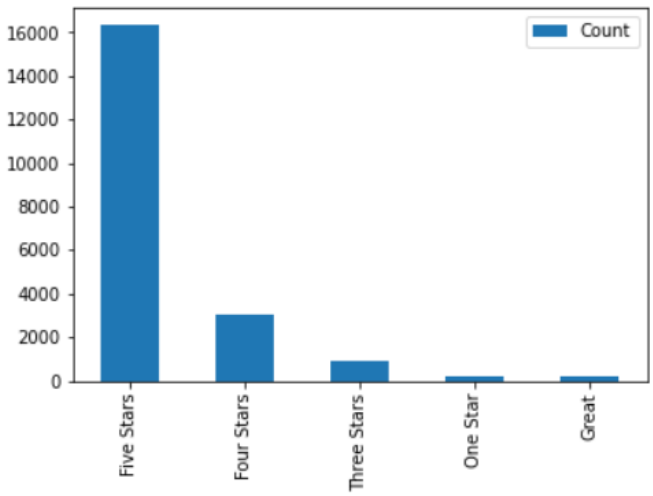


	Name	Count
0	good	412
1	great	277
2	ok	241
3	Great	241
4	works great	180

Summary :

This content of this column gives the summary of the reviews , it is the overview of reviews on few words .The analysis of summary column complements our analysis of reviewText that most of reviews are positive.

The most frequent words are :



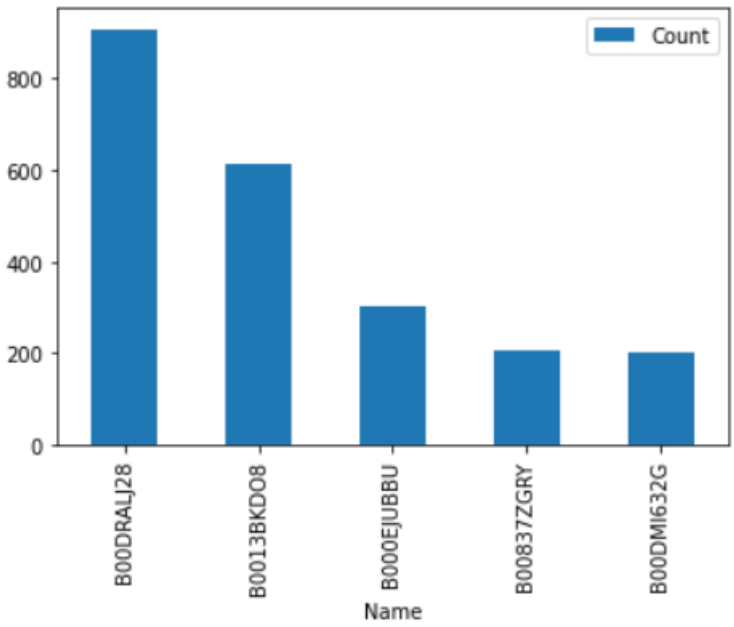
	Name	Count
0	Five Stars	16318
1	Four Stars	3033
2	Three Stars	900
3	One Star	233
4	Great	203

ProductID :

This column have the information about the id of the product who's review has been written by the customer.

There are **5105** unique products .

The most reviewed product id's are :

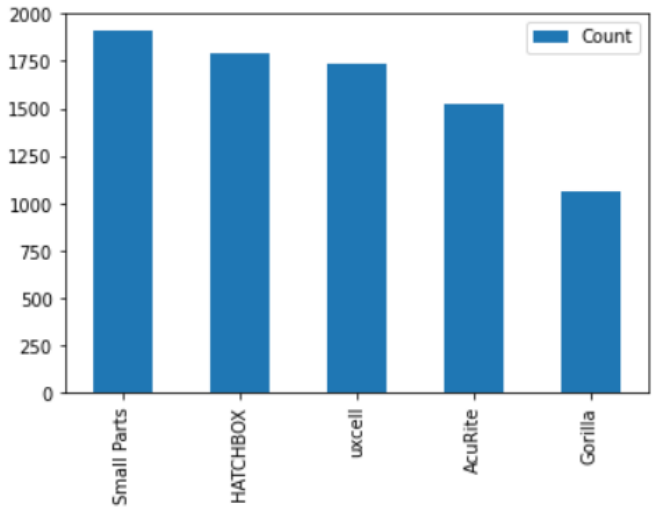


	Name	Count
0	B00DRALJ28	909
1	B0013BKDO8	614
2	B000EJUBBU	303
3	B00837ZGRY	204
4	B00DMI632G	202

Brand :

This tells the information about the brands in different category .We have **1842** unique brands.

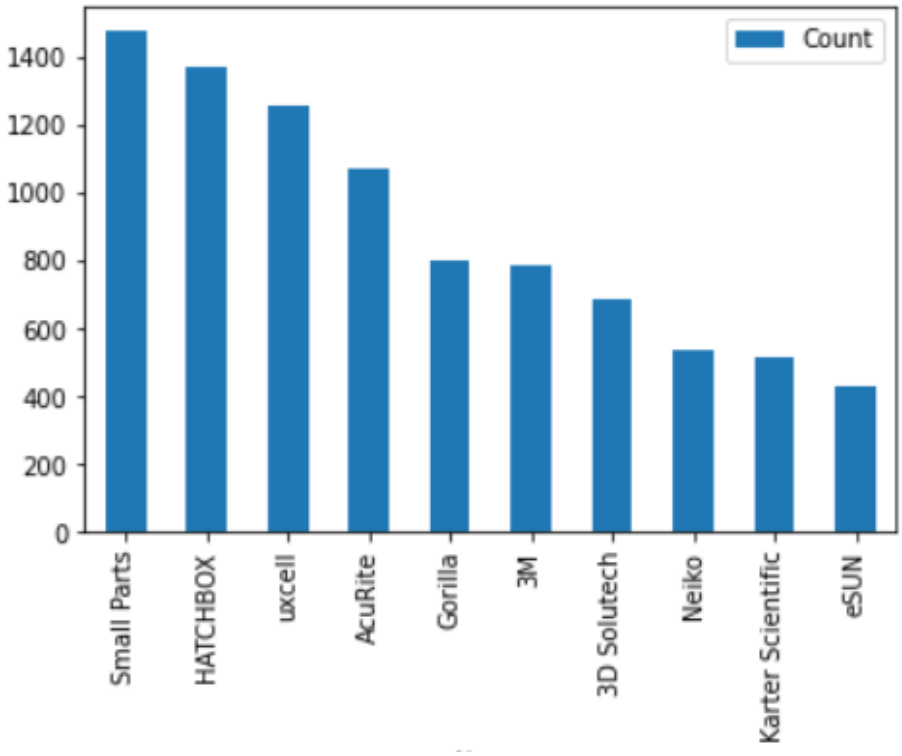
Most frequently reviewed brands are :



	Name	Count
0	Small Parts	1911
1	HATCHBOX	1788
2	uxcell	1738
3	AcuRite	1523
4	Gorilla	1061

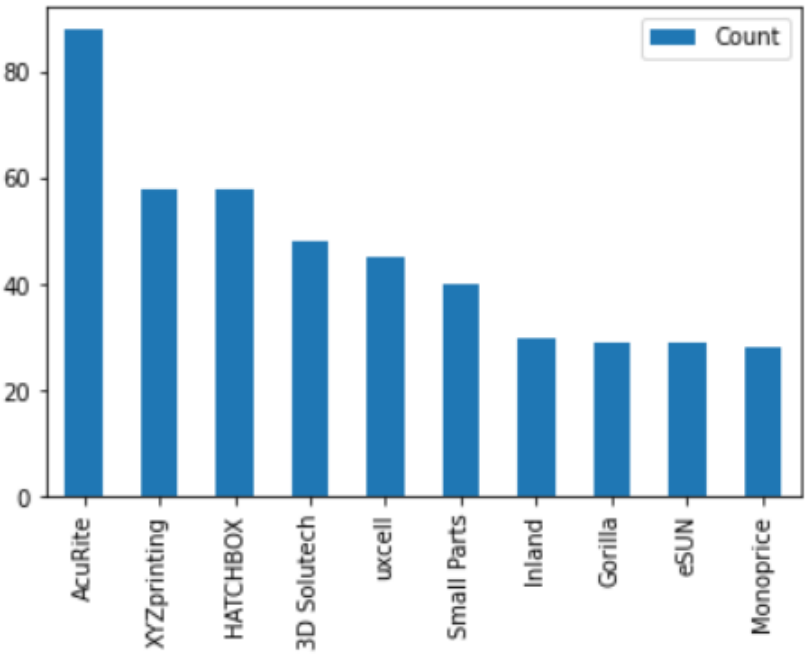
Top Brands with highest ratings :

These are the brands which are rating excellent by the customers and amazon should make these brands more visible so the customer get the good shopping experience.

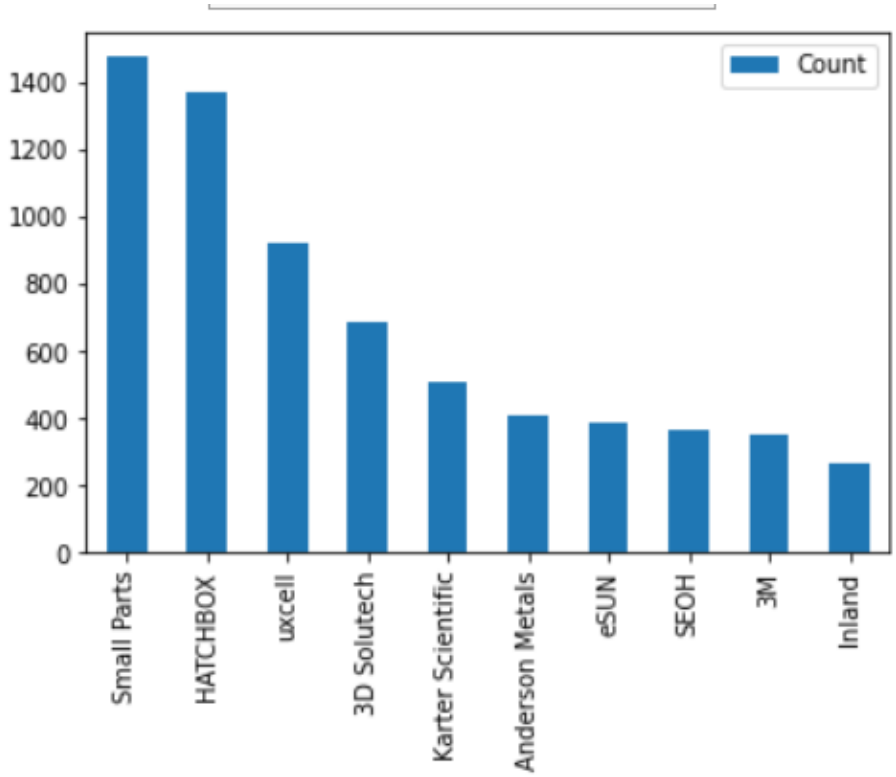


Brands with least ratings :

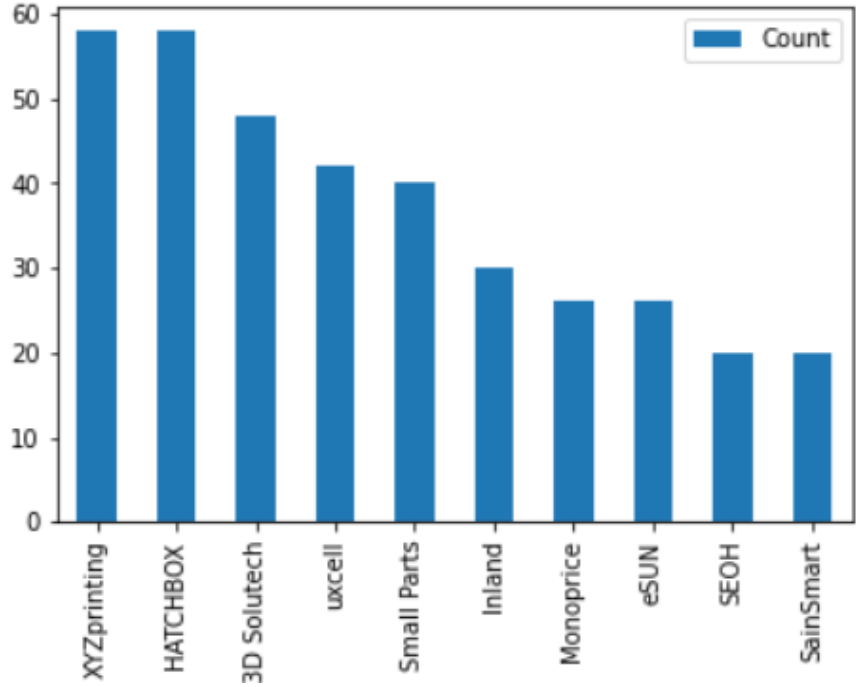
Amazon should make sure these barnds improve their quality where they have been rating negative so the customers experience remains good and helps in customer retention.



Top brands in Industrial and scientific category :

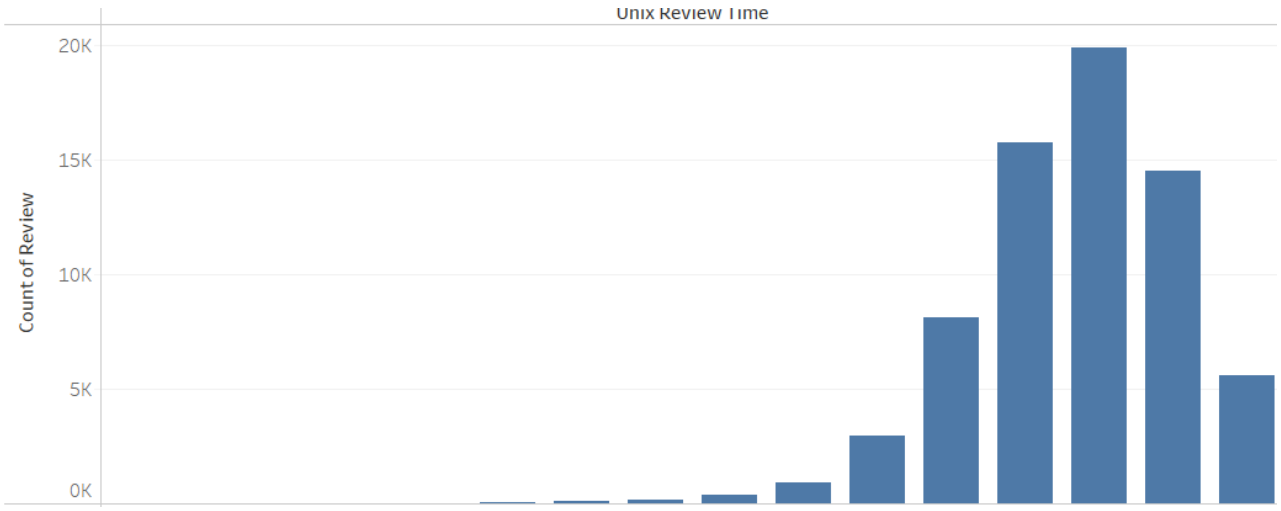


Worst Rated brands in Industrial and Scientific :

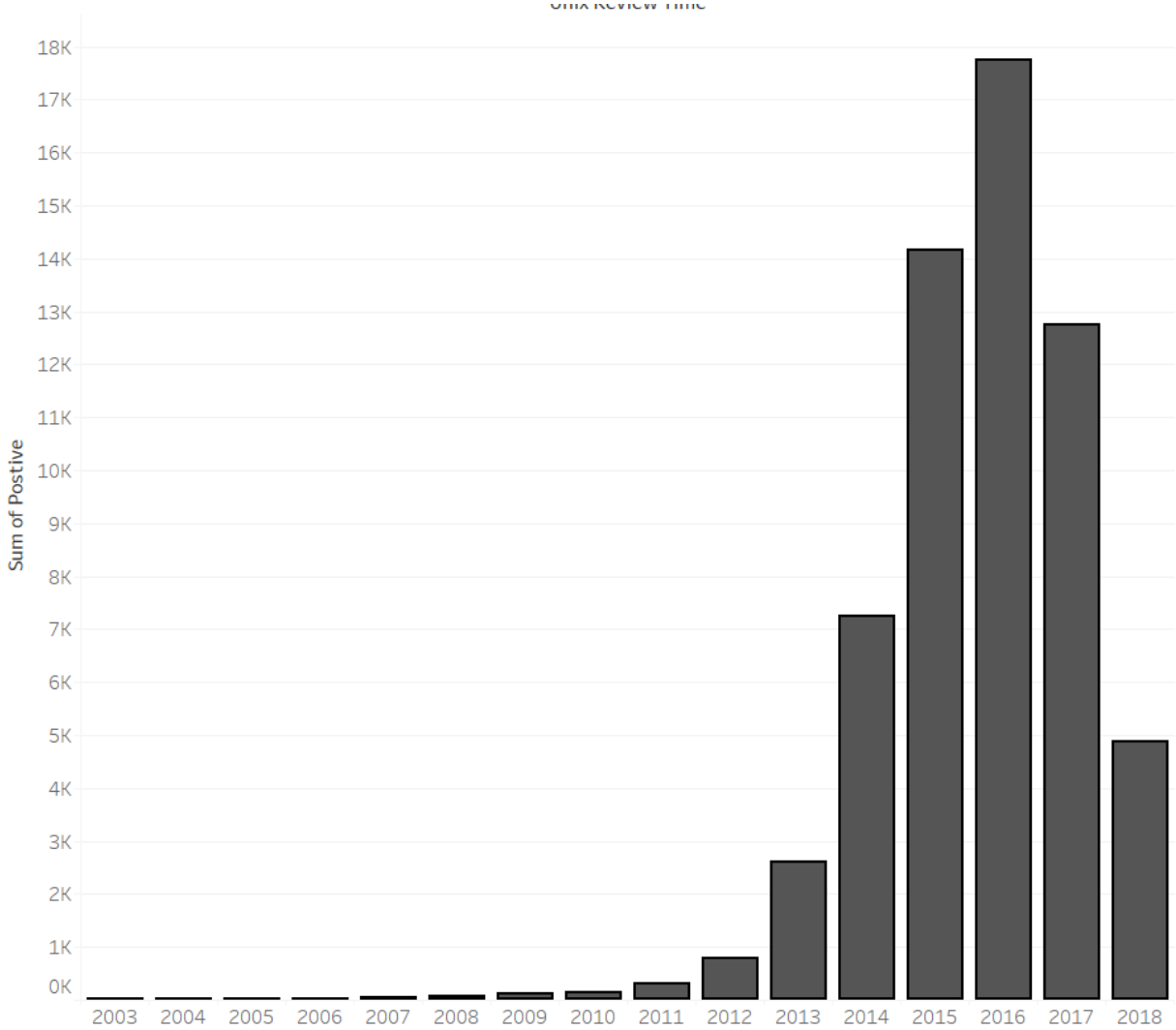


Review Count over years :

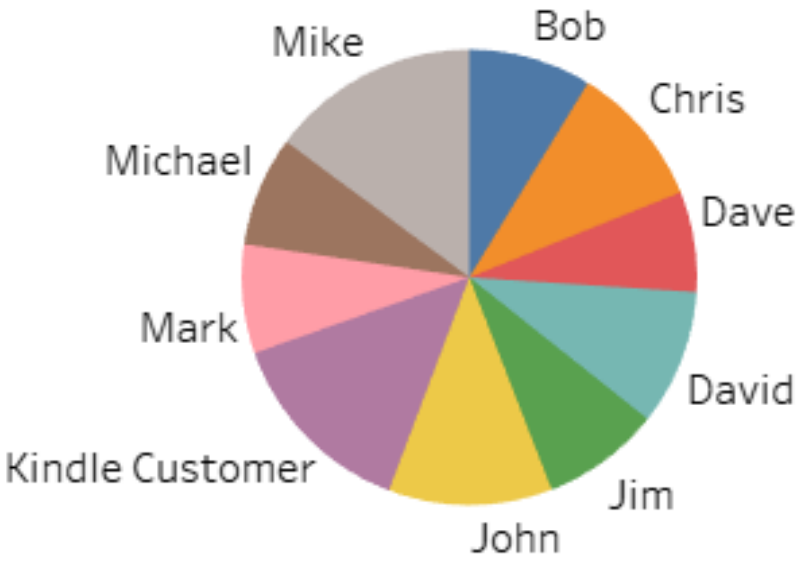
We see that the number of reviews over the year are decreasing over year.



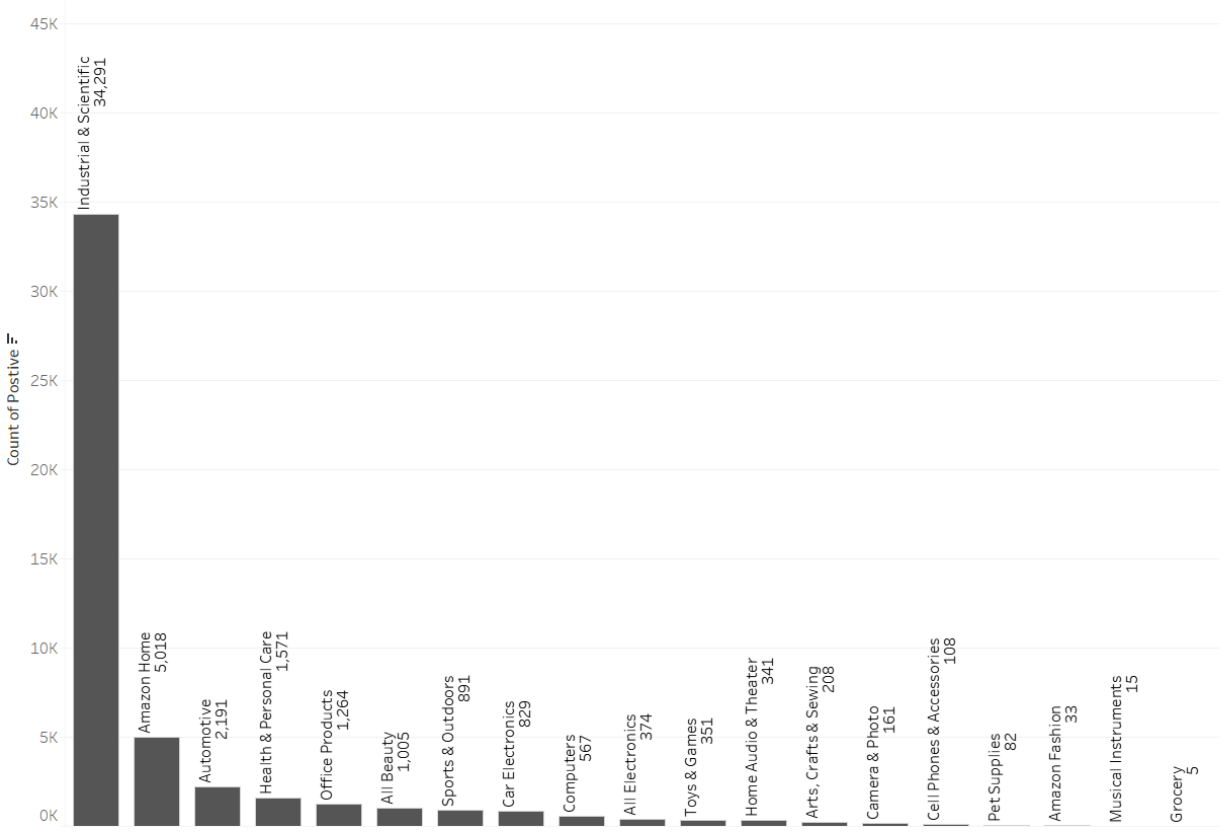
Positive review count over year :



Top Customers who gives positive review :



Categories with most Positive reviews :



Review count of different category over year :

Main Cat

Year of U...	All Ele..	Amazo..	Amazo..	Arts, C..	Autom..	Camer..	Car Ele..	Cell Ph..	Compu..	Grocery	Health..	Home ..	Indust..	Music..	Office ..	Pet Su..	Sports..	Tools ..	Toys & ..
2003													1						
2004																		4	
2005																		1	
2006			1												1			9	
2007			1										9		1			7	
2008			3		2								13					14	
2009		2	4		2						3		18		2		7	58	
2010	1	1	11		4						12	1	24	1	7		10	73	
2011		3	44		8					1	12	7	102		18		10	138	
2012	4	1	90	3	26		12		4	1	37	9	297		28		20	297	2
2013	15	2	240	8	112	2	42	17	8	2	100	19	1,300		47	6	65	871	13
2014	55	4	519	32	241	7	180	38	50	1	212	63	3,896	3	169	14	111	2,322	21
2015	155	10	935	41	508	14	314	24	135		392	97	7,851	8	268	26	218	4,547	53
2016	89	6	1,668	51	609	57	178	12	185		395	75	10,181	2	330	11	235	5,528	132
2017	42	4	1,084	51	522	65	84	15	119		297	55	7,603	1	292	14	151	3,892	99
2018	13		418	22	157	16	19	2	66		111	15	2,991		101	11	64	1,492	31

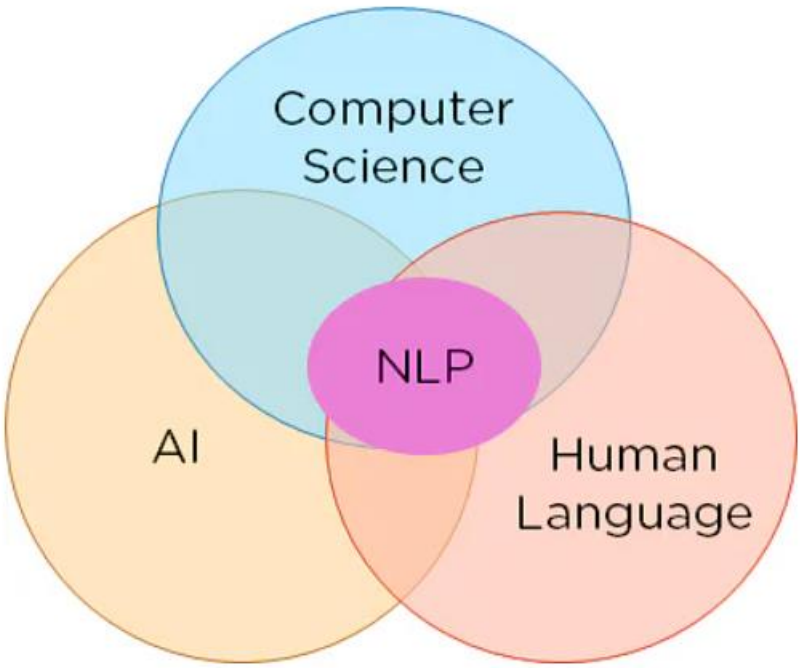
Sales of top brands over the years :

Unix Review Time

Brand	2003	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
3D Solutech										595	995	2,349	6,799	6,552
3M				21	48	22	119	303	1,842	1,437	2,841	4,013	2,535	804
AcuRite						31	71	82	245	795	1,559	6,756	4,504	1,478
eSUN										267	2,109	4,108	4,501	2,668
Gorilla	10	30	43	27	144	44	91	157	219	1,022	1,967	2,450	1,748	1,052
HATCHBOX										2,159	9,891	17,327	9,977	3,561
Karter Scientific							13	76	686	1,452	1,855	1,556	1,031	377
Neiko				37	119	40	57	480	1,349	2,386	3,590	4,776	3,356	1,035
Small Parts			105	167	137	8	230	596	1,787	4,129	8,411	7,998	4,348	1,677
uxcell							27	20	270	1,466	3,462	5,439	2,874	741

5. NLP (Natural Language Processing)

Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of artificial intelligence or AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.



5.1 Steps to perform NLP :

Segmentation :

You first need to break the entire document down into its constituent sentences. You can do this by segmenting the article along with its punctuations like full stops and commas.

E.g. . This is a great book; this book tells a lot about the history of India.

This is a great book

This book tells a lot about the history of India.

Tokenising:

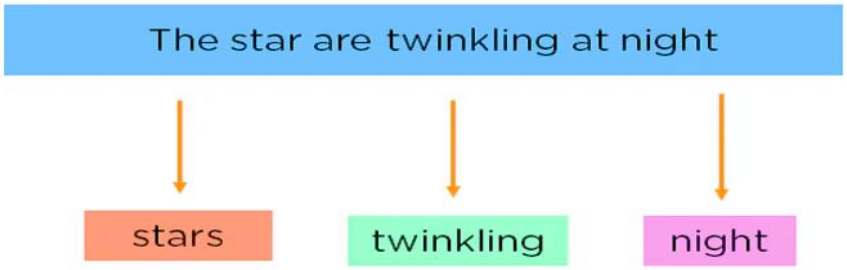
For the algorithm to understand these sentences, you need to get the words in a sentence and explain them individually to our algorithm. So, you break down your sentence into its constituent words and store them. This is called tokenizing, and each world is called a token

This is a great book

This | is | a | great | book

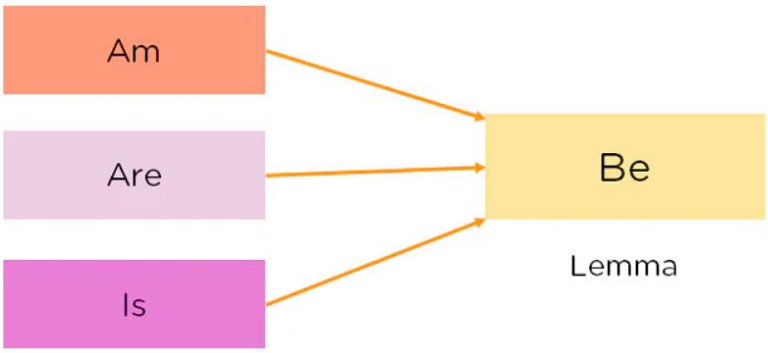
Removing Stop Words:

You can make the learning process faster by getting rid of non-essential words, which add little meaning to our statement and are just there to make our statement sound more cohesive. Stop words are available in abundance in any human language. By removing these words, we remove the low-level information from our text in order to give more focus to the important information. Words such as was, in, is, and, the, are called stop words and can be removed.



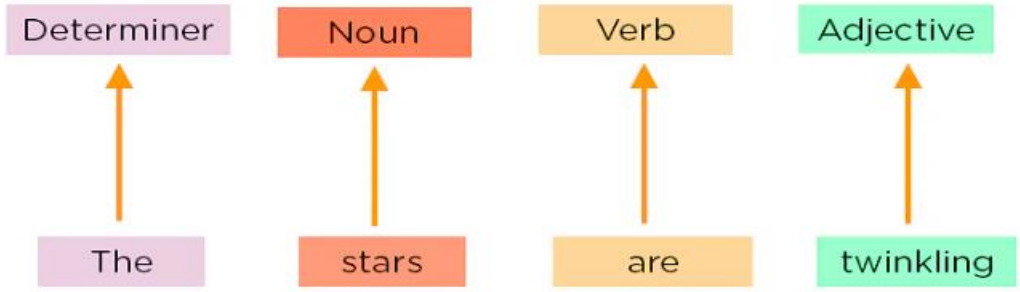
Stemming and Lemmatization:

The process of obtaining the Root Stem of a word. Root Stem gives the new base form of a word that is present in the dictionary and from which the word is derived. You can also identify the base words for different words based on the tense, mood, gender, etc. This is done to normalise text variables.



Part of Speech Tagging :

Every word have some part of speech , tagging word with pos is called Part of speech tagging to explain these concepts to our machine.



Named Entity Tagging:

Introducing machine to everyday names by flagging names of movies, important personalities or locations, etc that may occur in the document. You do this by classifying the words into subcategories. This helps you find any keywords in a sentence. The subcategories are person, location, monetary value, quantity, organization, movie.

5.2 Applications of NLP

- NLP is one of the ways that people have humanized machines and reduced the need for labour. It has led to the automation of speech-related tasks and human interaction. Some applications of NLP include:
- Translation Tools: Tools such as Google Translate, Amazon Translate, etc. translate sentences from one language to another using NLP.
- Chatbots: Chatbots can be found on most websites and are a way for companies to deal with common queries quickly.
- Virtual Assistants: Virtual Assistants like Siri, Cortana, Google Home, Alexa, etc can not only talk to you but understand commands given to them.
- Targeted Advertising: Have you ever talked about a product or service or just googled something and then started seeing ads for it? This is called targeted advertising, and it helps generate tons of revenue for sellers as they can reach niche audiences at the right time.
- Autocorrect: Autocorrect will automatically correct any spelling mistakes you make, apart from this grammar checkers also come into the picture which helps you write flawlessly.

6. SENTIMENT ANALYSIS:

Machine Learning models take numerical values as input. The reviews are made of sentences, so in order to extract patterns from the data; we need to find a way to represent it in a way that machine learning algorithm can understand, i.e. as a list of numbers. So machine can classify the revies into three segments – Positive, Negative, Neutral.

- After cleaning the review column i.e., removing stop words and normalizing textual data by making a user defined function named as Textclean.

- Here's how review look like after cleaning:

```
work purpose mark great durable affordable sandpaper pro grit cut fast
evenly random deep scratch like see cheap paper hint clog the adhesive
need permanent go con none happy performance
```

- So, we used the overall rating to map review as Positive, Negative or Neutral so we train our models accordingly.
- After cleaning our text we saved our dataframe file into csv file .

- Below mentioned is the count of the different types of review in our data:

```
Positive    60622
Negative     3974
Neutral      3965
Name: Sentiments, dtype: int64
```

- We see that our class distribution is highly imbalance and machine learning will not be efficient as machine will get trained for biasness towards positive class and our models will not work well with reviews for negative and neutral class . So we have to use **SMOTE** function to balance the data and make our model more accurate towards minority classes

6.1 PREPROCESSING:

1. Label Encoding:

Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

Example of label encoding: -

```
>>> le = preprocessing.LabelEncoder()
>>> le.fit(["paris", "paris", "tokyo", "amsterdam"])
LabelEncoder()
>>> list(le.classes_)
['amsterdam', 'paris', 'tokyo']
>>> le.transform(["tokyo", "tokyo", "paris"])
array([2, 2, 1]...)
>>> list(le.inverse_transform([2, 2, 1]))
['tokyo', 'tokyo', 'paris']
```

2. MinMaxScaler:

This estimator scales and translates each feature individually such that it is in the given range on the training set, e.g., between zero and one.

Example of MinMaxScaler: -

The transformation is given by:

```
X_std = (X - X.min(axis=0)) / (X.max(axis=0) - X.min(axis=0))
X_scaled = X_std * (max - min) + min
```

where min, max = feature_range.

This transformation is often used as an alternative to zero mean, unit variance scaling.

3. StandardScaler :

StandardScaler is a pre-processing method in machine learning that is used to normalize the features of a dataset. It scales the features of a dataset so that they have a mean of zero and a standard deviation of one. This is also known as standardizing the data. The standard score of a sample `x` is calculated as:

z = (x - u) / s samples

where `u` is the mean of the training samples `s` is the standard deviation of the training

```
from sklearn.preprocessing import StandardScaler

# X contains the feature data

# Create an instance of the StandardScaler
scaler = StandardScaler()

# Fit the scaler to the data
scaler.fit(X)

# Transform the data
X_scaled = scaler.transform(X)
```

SMOTE :

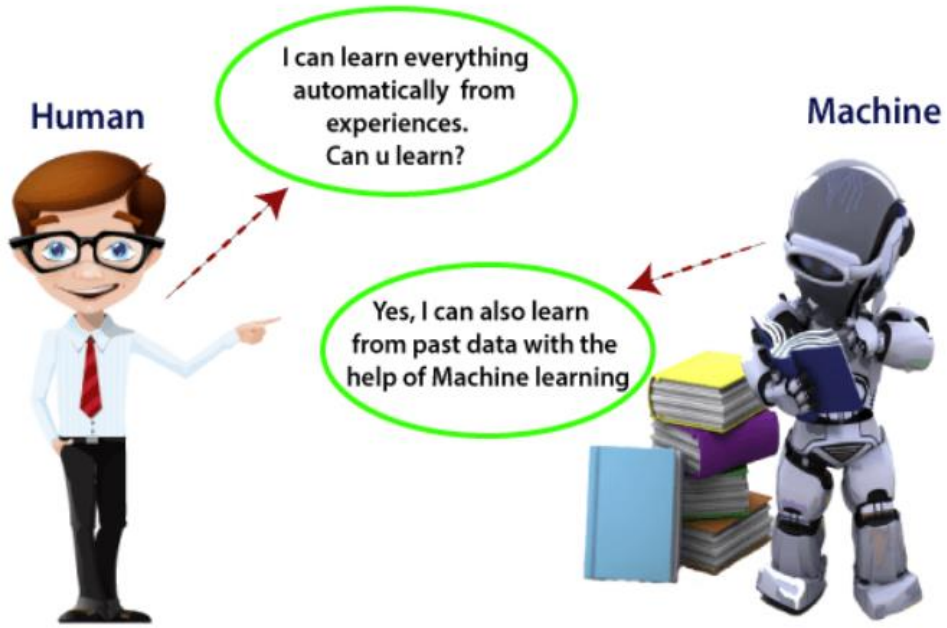
SMOTE stands for Synthetic Minority Oversampling Technique, it is a statistical technique for increas ing the number of cases in your dataset in a balanced way. The component works by generating new in stances from existing minority cases that you supply as input.

Because the classes are not equally spread thats why our model is not highly trained on other classes s o we will use the SMOTE technique to increase the samples of the class which have less data.

```
from imblearn.over_sampling import SMOTE

# X contains the feature data
# y contains the labels

# Create an instance of the SMOTE class
smote = SMOTE()

# Apply SMOTE to the data
X_resampled, y_resampled = smote.fit_resample(X, y)
```

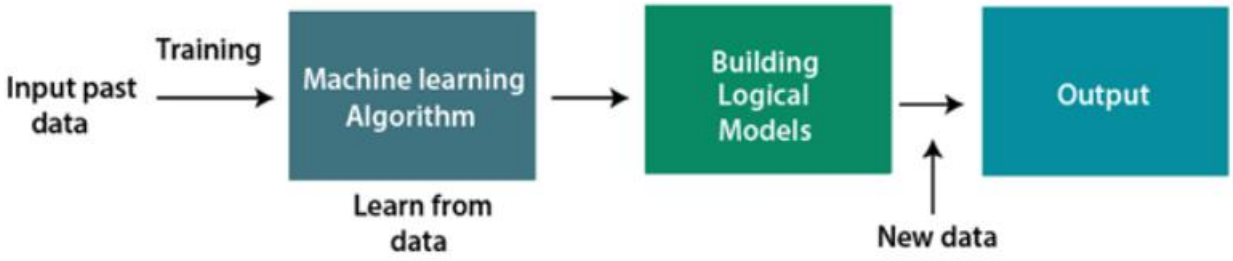
6.2 Machine Learning :

Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms for building mathematical models and making predictions using historical data or information. Currently, it is being used for various tasks such as image recognition, speech recognition, email filtering, Facebook auto-tagging, recommender system, and many more.



Machine Learning is a subset of **artificial intelligence** that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own.

A Machine Learning system **learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it**. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.



Classification of Machine Learning

Supervised learning

Unsupervised learning

Reinforcement learning

1) Supervised Learning

Supervised learning is a type of machine learning method in which we provide sample labelled data to the machine learning system in order to train it, and on that basis, it predicts the output.

The system creates a model using labelled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is **spam filtering**.

Supervised learning can be grouped further in two categories of algorithms:

- **Classification**
- **Regression**

2) Unsupervised Learning

Unsupervised learning is a learning method in which a machine learns without any supervision.

The training is provided to the machine with the set of data that has not been labelled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns.

In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data. It can be further classifieds into two categories of algorithms:

- **Clustering**
- **Association**

3) Reinforcement Learning

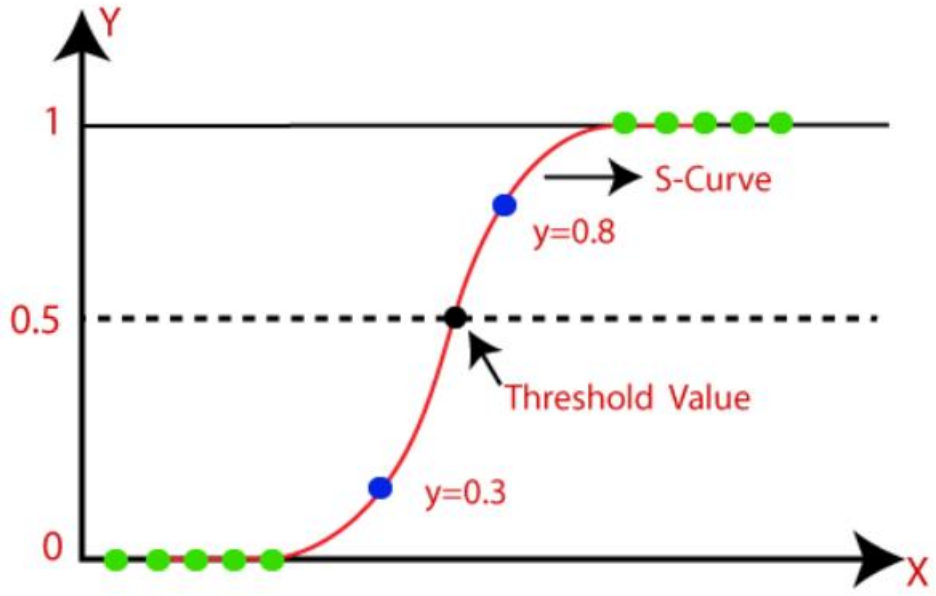
Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improves its performance.

The robotic dog, which automatically learns the movement of his arms, is an example of Reinforcement learning.

6.3 Different Machine Learning Models used for sentiment classifications are :

6.3.1 LogisticRegression:

It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1 We used this ML model with **OneVsRest** Classifier because we had 3 classes to classify. Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:



6.3.2 MultinomialNB:

The Multinomial Naive Bayes algorithm is a Bayesian learning approach popular in Natural Language Processing (NLP). The program guesses the tag of a text, such as an email or a newspaper story, using the Bayes theorem. It calculates each tag's likelihood for a given sample and outputs the tag with the g reatest chance.

6.3.3 RandomForestClassifier:

The Random Forest classifier creates a set of decision trees from a randomly selected subset of the training set. It is basically a set of decision trees (DT) from a randomly selected subset of the training set and then It collects the votes from different decision trees to decide the final prediction. A random forest is a meta estimator that fits a number of decision tree classifiers on various Sub samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting

6.3.4 AdaBoostClassifier:

An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

6.4 Evaluation Metrics:

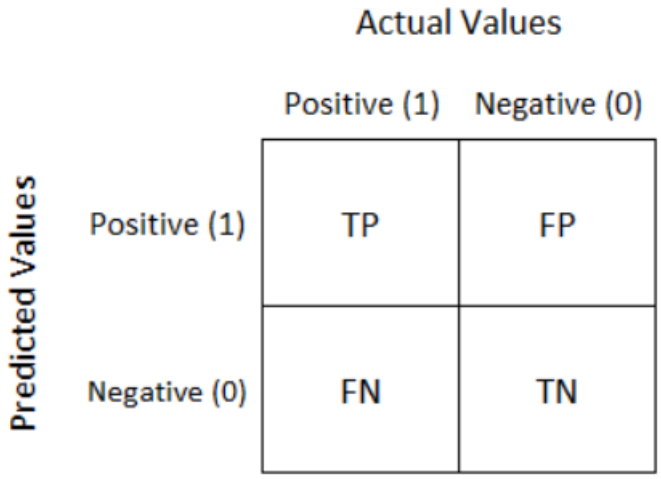
Evaluating a model is a core part of building an effective machine learning model

There are several evaluation metrics in machine learning, like confusion matrix, classification report etc

Evaluation metrics explain the performance of a model. An important aspect of evaluation metrics is their capability to discriminate among model results.

6.4.1 Confusion Matrix :-

A confusion matrix is a table that is often used to evaluate the performance of a classification algorithm. It is a table with two rows and two columns that reports the number of true positives, false positives, true negatives, and false negatives.



TP – True Positive, it's a value which is True and Predicted True

FP - False Positive, it's a value which is False and Predicted True

TN - True Negative, it's a value which is False and Predicted False

FN - False Negative, it's a value which is True and Predicted False

- On the basis of this confusion matrix, we are able to calculate following parameters:

1. **Accuracy**: The proportion of the total number of predictions that were correct.

$$Accuracy = \frac{Correct\ prediction}{Total\ cases} * 100\%$$

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} * 100\%$$

2. **Precision:** The proportion of positive cases that were correctly identified.

$$Precision = \frac{TP}{TP + FP}$$

3. **Recall** : The proportion of actual positive cases which are correctly identified.

$$Recall = \frac{TP}{TP + FN}$$

4. **F1-Score:.** It is primarily used to compare the performance of two classifiers. Suppose that classifier It is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1]. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances). The greater the F1 Score, the better is the performance of our model.

F-Measure = (2 * Precision * Recall) / (Precision + Recall)

6.4.2 Classification report :

The classification report is a way to evaluate the performance of a classification algorithm. It is a summary of a model's performance that includes the precision, recall, f1-score, and support for each class.

	precision	recall	f1-score	support
0	0.99	0.97	0.98	18573
1	0.96	0.96	0.96	18147
2	0.94	0.96	0.95	17840
accuracy			0.96	54560
macro avg	0.96	0.96	0.96	54560
weighted avg	0.96	0.96	0.96	54560

6.5 Techniques used to convert words into numerical form:

6.5.1 TFIDF Vectorizer (Term Frequency Inverse Document Frequency)

Term Frequency - Inverse Document Frequency (TF-IDF) is a widely used statistical method in natural language processing and information retrieval. It measures how important a term is within a document relative to a collection of documents (i.e., relative to a corpus). Words within a text document are transformed into importance numbers by a text vectorization process. There are many different text vectorizations scoring schemes, with TF-IDF being one of the most common.

As its name implies, TF-IDF vectorizes/scores a word by multiplying the word's Term Frequency (TF) with the Inverse Document Frequency (IDF).

Term Frequency: TF of a term or word is the number of times the term appears in a document compared to the total number of words in the document.

$$TF = \frac{\text{number of times the term appears in the document}}{\text{total number of terms in the document}}$$

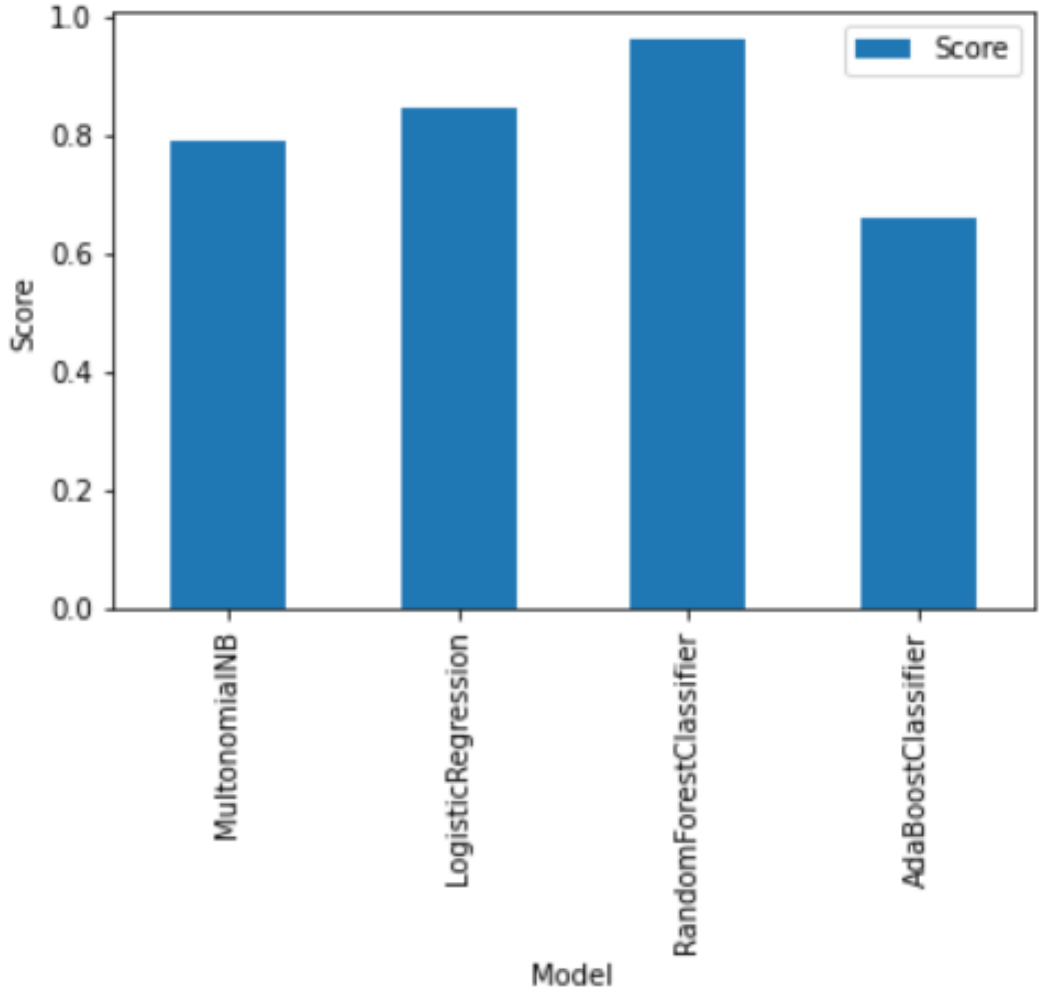
Inverse Document Frequency: IDF of a term reflects the proportion of documents in the corpus that contain the term. Words unique to a small percentage of documents (e.g., technical jargon terms) receive higher importance values than words common across all documents (e.g., a, the, and)

$$IDF = log(\frac{\text{number of the documents in the corpus}}{\text{number of documents in the corpus contain the term}})$$

The TF-IDF of a term is calculated by multiplying TF and IDF scores.

$$TF\text{-}IDF = TF * IDF$$

We applied 4 Models in this Technique and below charts tells the accuracy score of all the models :



Accuracy score of RandomForestClassifier is higher but the model have been overfitted because it's not able to recognize sentiments correctly on unknown data

Logistic Regression turned out to be the best model because of its ability to recognize pattern on unknown data and giving good results for every class..

Detailed Comparison of Models :

TFIDF	AdaBoostClassifier	0	0.665761	0.728373	0.695661
		1	0.605341	0.578962	0.591857
		2	0.715859	0.679829	0.697379
		accuracy	0.662482	0.662482	0.662482
		macro avg	0.662320	0.662388	0.661633
		weighted avg	0.662451	0.662482	0.661745
	LogisticRegression	0	0.873081	0.926315	0.898911
		1	0.802858	0.821281	0.811965
		2	0.859098	0.787393	0.821684
		accuracy	0.844923	0.844923	0.844923
		macro avg	0.845012	0.844996	0.844186
		weighted avg	0.845068	0.844923	0.844176
	MultinomialNB	0	0.813148	0.859014	0.835452
		1	0.744050	0.763914	0.753851
		2	0.817851	0.750260	0.782599
		accuracy	0.791019	0.791019	0.791019
		macro avg	0.791683	0.791063	0.790634
		weighted avg	0.791762	0.791019	0.790650
	RandomForestClassifier	0	0.968826	0.990205	0.979399
		1	0.960710	0.961664	0.961186
		2	0.957791	0.935812	0.946674
		accuracy	0.962518	0.962518	0.962518

Hyperparameter Tuning of our best model :

A Machine Learning model is defined as a mathematical model with a number of parameters that need to be learned from the data. By training a model with existing data, we are able to fit the model parameters.

However, there is another kind of parameter, known as Hyperparameters, that cannot be directly learned fro m the regular training process. They are usually fixed before the actual training process begins. These param eters express important properties of the model such as its complexity or how fast it should learn.

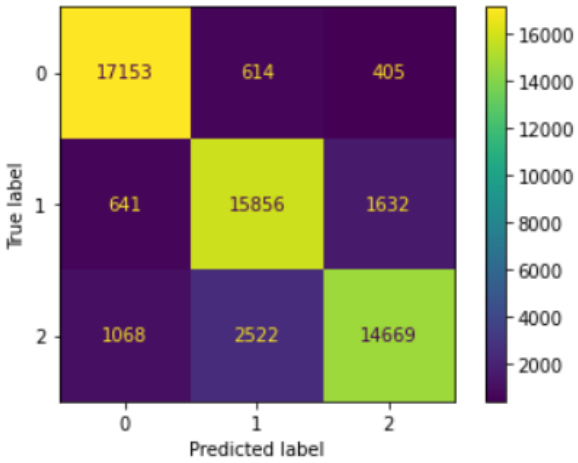
Parameters of our models:-

```
dict_keys(['estimator__C', 'estimator__class_weight', 'estimator__dual',
'estimator__fit_intercept', 'estimator__intercept_scaling', 'estimator__l1_ratio',
'estimator__max_iter', 'estimator__multi_class', 'estimator__n_jobs',
'estimator__penalty', 'estimator__random_state', 'estimator__solver',
'estimator__tol', 'estimator__verbose', 'estimator__warm_start', 'estimator',
'n_jobs'])
```

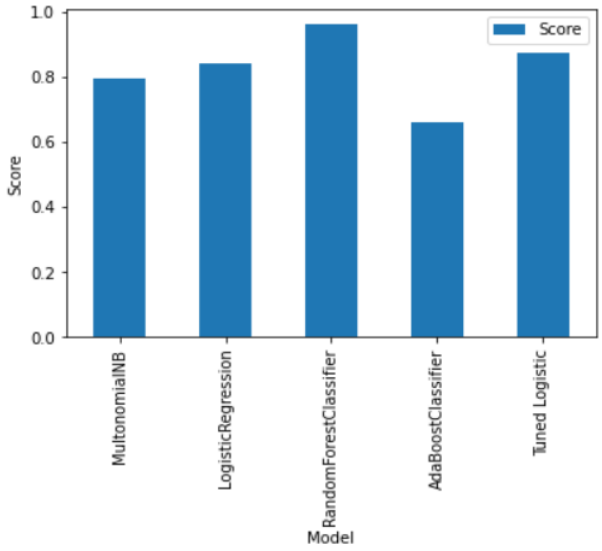
After tuning these parameters we were able to see nearby 4% increase in accuracy

Here is the classification report of our best model with this technique.

	precision	recall	f1-score	support
0	0.94	0.91	0.93	18862
1	0.87	0.83	0.85	18992
2	0.80	0.88	0.84	16706
accuracy			0.87	54560
macro avg	0.87	0.87	0.87	54560
weighted avg	0.88	0.87	0.87	54560

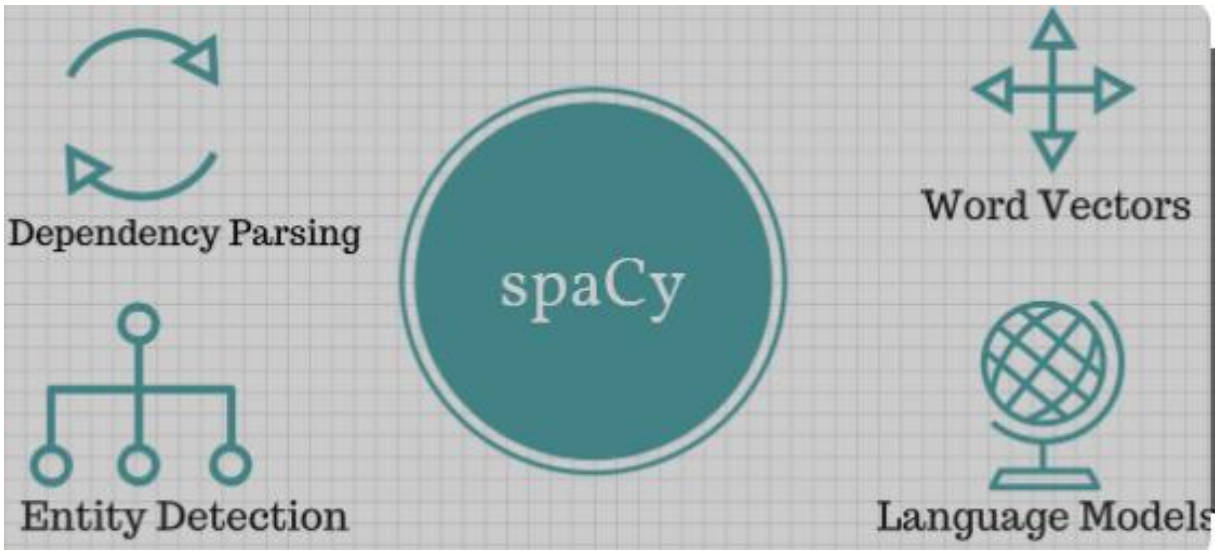


This Model is working well for every class.



6.5.2 Spacy Vectorization :

spaCy is a free, opensource library for NLP in Python written in Cython. spaCy is designed to make it easy to build systems for information extraction or general-purpose natural language processing.



Reviews were converted into vector to fit into the models using spacy vectors , shape of each vector is 300.

Performance of two models is showcased below :

			precision	recall	f1-score
Technique	Model	index			
Spacy	LogisticRegression	0	0.234024	0.715899	0.352740
		1	0.108415	0.489726	0.177529
		2	0.975280	0.629096	0.764839
		accuracy	0.626039	0.626039	0.626039
		macro avg	0.439240	0.611574	0.431703
		weighted avg	0.884577	0.626039	0.708429
	RandomForestClassifier	0	0.735537	0.077324	0.139937
		1	0.199013	0.103596	0.136261
		2	0.894456	0.972384	0.931793
		accuracy	0.872964	0.872964	0.872964
		macro avg	0.609669	0.384435	0.402664
		weighted avg	0.846073	0.872964	0.842309

These models are working great for the positive classes but because there is a class imbalance that's why for other classes performance is poor and SMOTE did not work on the vector.

Example of vector :

```
array([ 1.8963e-01, -4.0309e-01,  3.5350e-01, -4.7907e-01, -4.3311e-01,
        2.3857e-01,  2.6962e-01,  6.4332e-02,  3.0767e-01,  1.3712e+00,
       -3.7582e-01, -2.2713e-01, -3.5657e-01, -2.5355e-01,  1.7543e-02,
        3.3962e-01,  7.4723e-02,  5.1226e-01, -3.9759e-01,  5.1333e-03,
       -3.0929e-01,  4.8911e-02, -1.8610e-01, -4.1702e-01, -8.1639e-01,
       -1.6908e-01, -2.6246e-01, -1.5983e-02,  1.2479e-01, -3.7276e-02,
       -5.7125e-01, -1.6296e-01,  1.2376e-01, -5.5464e-02,  1.3244e-01,
        2.7519e-02,  1.2592e-01, -3.2722e-01, -4.9165e-01, -3.5559e-01,
       -3.0630e-01,  6.1185e-02, -1.6932e-01, -6.2405e-02,  6.5763e-01,
       -2.7925e-01, -3.0450e-03, -2.2400e-02, -2.8015e-01, -2.1975e-01,
       -4.3188e-01,  3.9864e-02, -2.2102e-01, -4.2693e-02,  5.2748e-02,
        2.8726e-01,  1.2315e-01, -2.8662e-02,  7.8294e-02,  4.6754e-01,
       -2.4589e-01, -1.1064e-01,  7.2250e-02, -9.4980e-02, -2.7548e-01,
       -5.4097e-01,  1.2823e-01, -8.2408e-02,  3.1035e-01, -6.3394e-02,
       -7.3755e-01, -5.4992e-01,  9.9999e-02, -2.0758e-01, -3.9674e-02,
        2.0664e-01, -9.7557e-02, -3.7092e-01,  2.7901e-01, -6.2218e-01,
       -1.0280e-01,  2.3271e-01,  4.3838e-01,  3.2445e-02, -2.9866e-01,
       -7.3611e-02,  7.1594e-01,  1.4241e-01,  2.7770e-01, -3.9892e-01,
        3.6656e-02,  1.5759e-01,  8.2014e-02, -5.7343e-01,  3.5457e-01,
        2.2491e-01, -6.2699e-01, -8.8106e-02,  2.4361e-01,  3.8533e-01,
       -1.4083e-01,  1.7691e-01,  7.0897e-02,  1.7951e-01, -4.5907e-01,
       -8.2120e-01, -2.6631e-02,  6.2549e-02,  4.2415e-01, -8.9630e-02,
       -2.4654e-01,  1.4156e-01,  4.0187e-01, -4.1232e-01,  8.4516e-02,
       -1.0626e-01,  7.3145e-01,  1.9217e-01,  1.4240e-01,  2.8511e-01,
       -2.9454e-01, -2.1948e-01,  9.0460e-01, -1.9098e-01, -1.0340e+00,
       -1.5754e-01, -1.1964e-01,  4.9888e-01, -1.0624e+00, -3.2820e-01,
       -1.1232e-02, -7.9482e-01,  3.7275e-01, -6.8710e-03, -2.5772e-01,
       -4.7005e-01, -4.1387e-01, -6.4089e-02, -2.8033e-01, -4.0778e-02,
       -2.4866e+00,  6.2494e-03, -1.0210e-02,  1.2752e-01,  3.4965e-01,
       -1.2571e-01,  3.1570e-01,  4.1926e-01,  2.0056e-01, -5.5984e-01,
       -2.2801e-01,  1.2012e-01, -2.0518e-03, -8.9764e-02, -8.0373e-02,
        1.1969e-02, -2.6978e-01,  3.4829e-01,  7.3664e-03, -1.1137e-01,
        6.3410e-01,  3.8449e-01, -6.2248e-01,  4.1145e-02,  2.5922e-01,
        6.5811e-01, -4.9548e-01, -1.3030e-01, -3.8279e-01,  1.1156e-01,
       -4.3085e-01,  3.4473e-01,  2.7109e-02, -2.5108e-01, -2.8011e-01,
        2.1662e-01,  3.2660e-01,  5.5895e-02,  7.6077e-02, -5.2480e-02,
```

6.6 Fasttext :

FastText is an opensource, free library from Facebook AI Research(FAIR) for learning word embeddi ngs and word classifications. This model allows creating unsupervised learning or supervised learning algorithm for obtaining vector representations for words. It also evaluates these models.

Evaluation of Fasttext model :

```
No. of Test Samples: 13713
Precision Score: 0.9084810034274047
Recall: 0.9084810034274047
F1 score: 0.9084810034274047
```

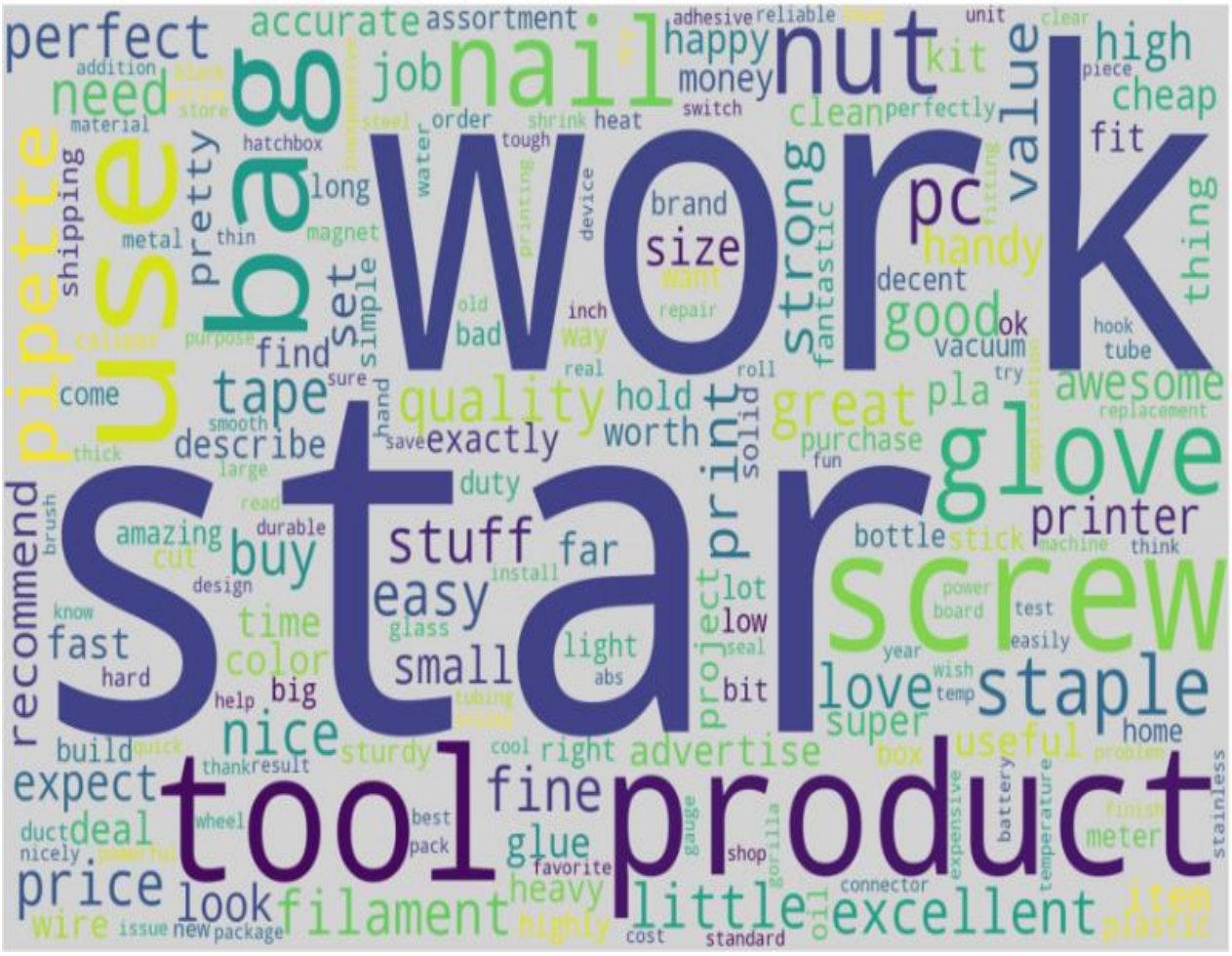
We also created a manual user defined function named as raw_analyzer to classify sentiments.

This works on counting the count of positive, negative and neutral in the review.

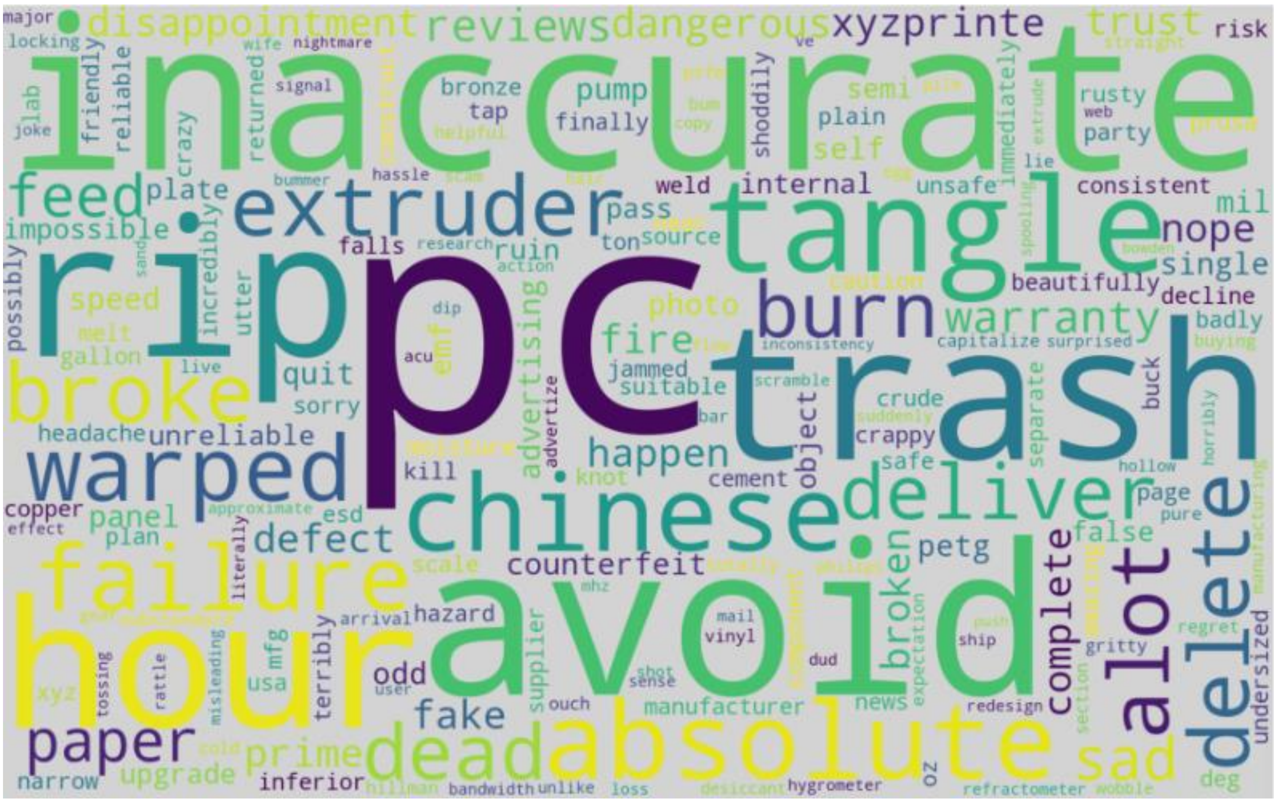
6.7 Conclusion:

We are taking hyper tuned Logistic Regression with TFIDF technique as our final model because it is working well on unknown reviews very well.

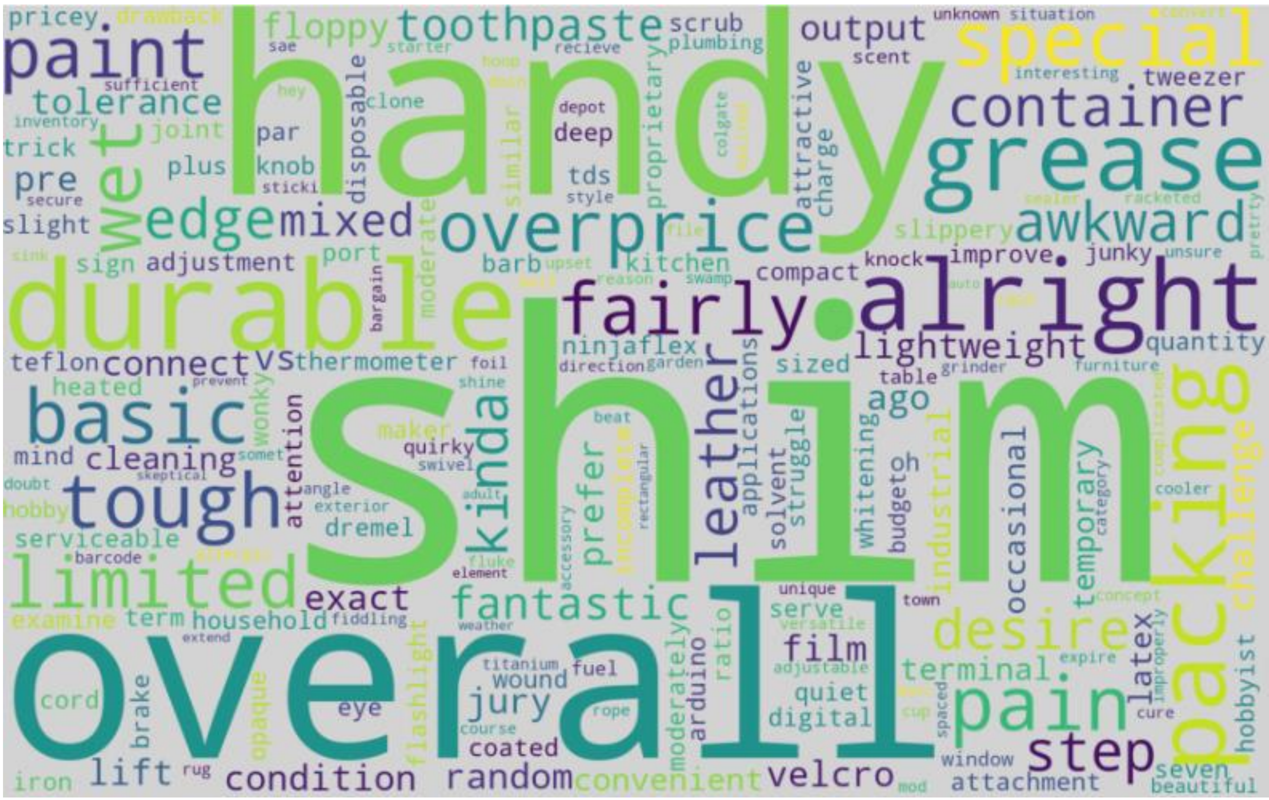
Positive Words:



Negative Words:



Neutral Words:

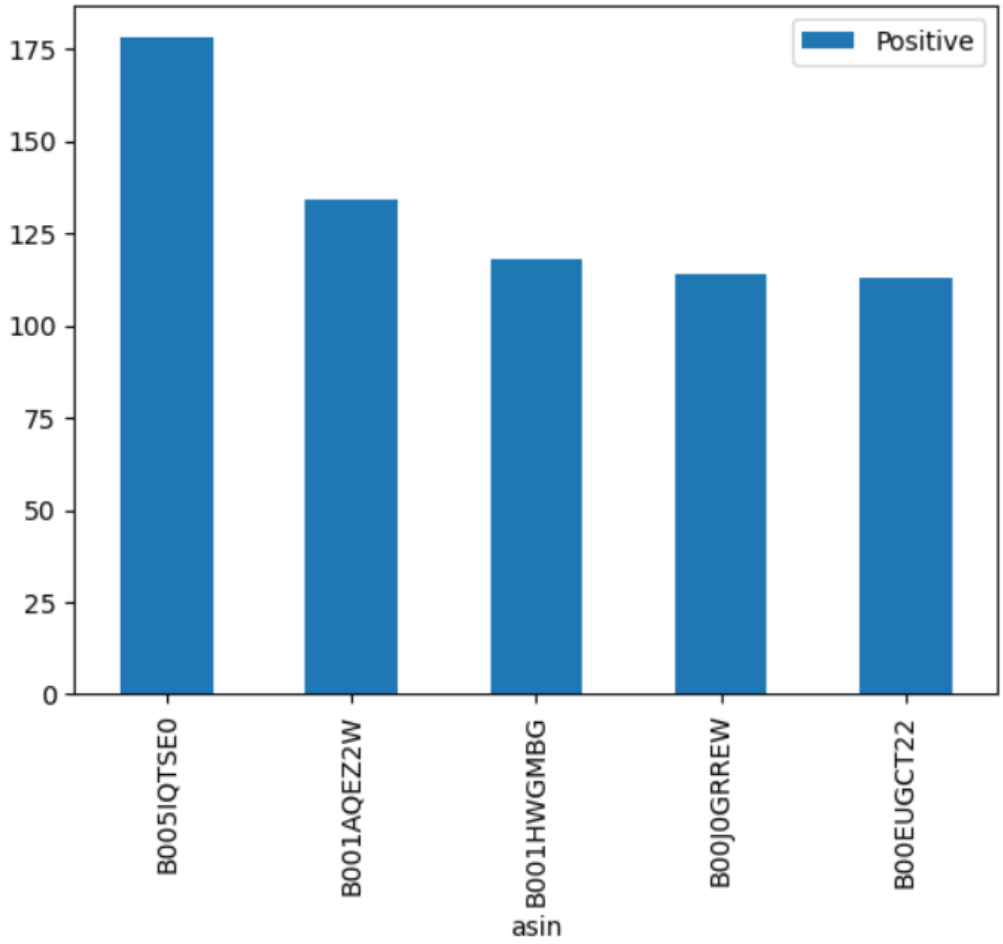


There is a translator given if the reviews come in a different language, so our translator can work and give the sentiment of the review.

There is also one search recommendation function to help you choose best products according to your need.

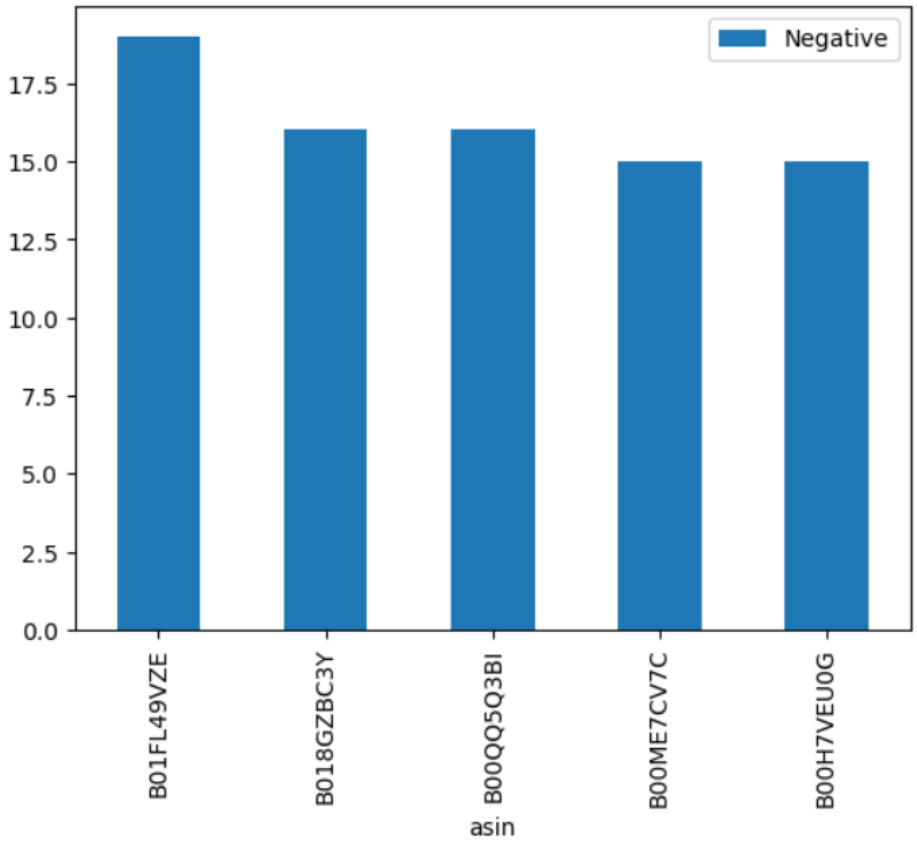
Recommended Products:

These are the recommend products on the basis of positive reviews and customer satisfaction.



	asin	Total_Sold	Negative	Neutral	Positive	Price	Sales	Rank	Verified_True	Verified_False	Main_Cat	Brand	Title
1989	B005IQTSE0	184	2	8	178	5.49	1032.12	3222.0	184	4	Industrial & Scientific	Pipettes	Plastic Transfer Pipettes 3ml, Graduated, Pack...
1156	B001AQEZ2W	139	5	5	134	28.95	4168.80	644.0	139	5	Industrial & Scientific	iGaging	Industrial & Scientific" />
1257	B001HWGMBG	121	6	5	118	7.21	930.09	1013.0	121	8	Industrial & Scientific	Precision Brand	Precision Brand M6S Micro Seal, Miniature All ...
3474	B00J0GRREW	117	2	2	114	19.99	2358.82	309.0	117	1	Industrial & Scientific	HATCHBOX	HATCHBOX PLA 3D Printer Filament, Dimensional ...
3114	B00EUGCT22	126	5	9	113	5.89	748.03	456.0	126	1	Industrial & Scientific	Gorilla	Gorilla Crystal Clear Duct Tape, 1.88” x...

Bad Products:



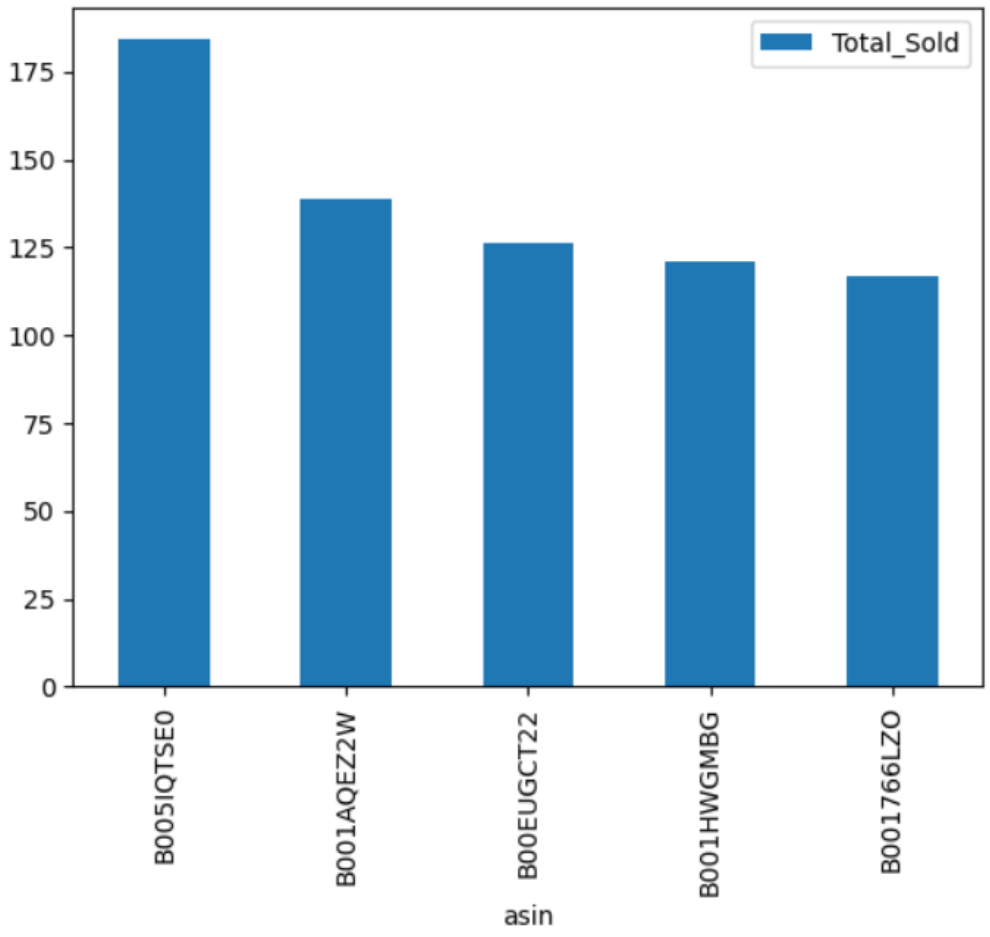
	asin	Total_Sold	Negative	Neutral	Positive	Price	Sales	Rank	Verified_True	Verified_False	Main_Cat	Brand	Title
5023	B01FL49VZE	98	19	12	78	189.90	20699.10	2645.0	98	11	Industrial & Scientific	Monoprice	Monoprice Select Mini 3D Printer v2 - White Wi...
4654	B018GZBC3Y	66	16	4	51	252.90	17955.90	3648.0	66	5	Industrial & Scientific	Monoprice	Monoprice Maker Select 3D Printer v2 With Larg...
3970	B00QQ5Q3BI	70	16	9	46	7.99	567.29	4815.0	70	1	Industrial & Scientific	Signstek	Signstek 3D Printer MK2 MK3 Heated Bed Tempere...
3717	B00ME7CV7C	95	15	2	82	17.99	1781.01	732.0	95	4	Industrial & Scientific	3D Solutech	3D Solutech Real Black 3D Printer PLA Filament...
3313	B00H7VEU0G	33	15	5	18	523.48	19892.24	143359.0	33	5	Industrial & Scientific	XYZprinting	XYZprinting Da Vinci 1.0 3D Printer, Grey

Neutral Products:

	asin	Total_Sold	Negative	Neutral	Positive	Price	Sales	Rank	Verified_True	Verified_False	Main_Cat	Brand
5023	B01FL49VZE	98	19	12	78	189.90	20699.10	2645.0	98	11	Industrial & Scientific	Monoprice
423	B000FMWU42	86	3	10	75	7.63	671.44	2072.0	86	2	Industrial & Scientific	Small Parts
431	B000FN1FLA	84	3	10	73	7.63	656.18	91651.0	84	2	Industrial & Scientific	Small Parts
3970	B00QQ5Q3BI	70	16	9	46	7.99	567.29	4815.0	70	1	Industrial & Scientific	Signstek
3114	B00EUGCT22	126	5	9	113	5.89	748.03	456.0	126	1	Industrial & Scientific	Gorilla

Most Selling product with positive reviews:

	asin	Total_Sold	Negative	Neutral	Positive	Price	Sales	Rank	Verified_True	Verified_False	Main_Cat	Brand
1989	B005IQTSE0	184	2	8	178	5.49	1032.12	3222.0	184	4	Industrial & Scientific	Pipettes
1156	B001AQEZ2W	139	5	5	134	28.95	4168.80	644.0	139	5	Industrial & Scientific	iGaging
3114	B00EUGCT22	126	5	9	113	5.89	748.03	456.0	126	1	Industrial & Scientific	Gorilla
1257	B001HWGMBG	121	6	5	118	7.21	930.09	1013.0	121	8	Industrial & Scientific	Precision Brand
1080	B001766LZO	117	9	4	109	7.85	957.70	65587.0	117	5	Industrial & Scientific	American Terminal
3474	B00J0GRREW	117	2	2	114	19.99	2358.82	309.0	117	1	Industrial & Scientific	HATCHBOX
3912	B00OZSL8UE	117	3	3	113	6.99	831.81	1935.0	117	2	Industrial & Scientific	URBEST
3479	B00J0H8EWA	115	12	6	98	19.99	2318.84	458.0	115	1	Industrial & Scientific	HATCHBOX
3478	B00J0H6NNM	109	7	4	100	19.99	2218.89	813.0	109	2	Industrial & Scientific	HATCHBOX
1113	B0018ACR6G	107	1	4	110	10.62	1221.30	1221.0	107	8	Industrial & Scientific	Dykem
4300	B01092XXD4	104	5	6	97	34.99	3778.92	220.0	104	4	Industrial & Scientific	HATCHBOX

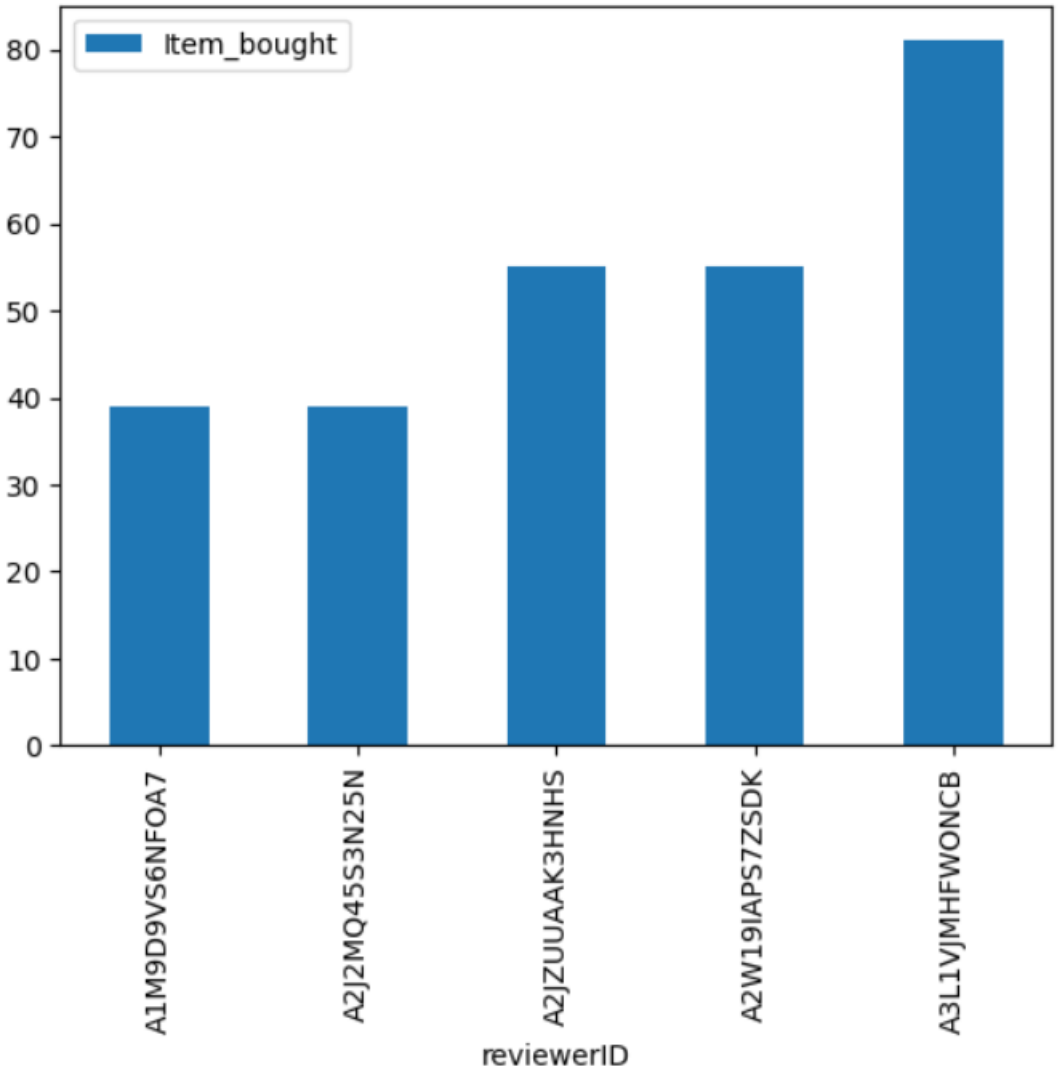


Customers:

Most loyal customers:

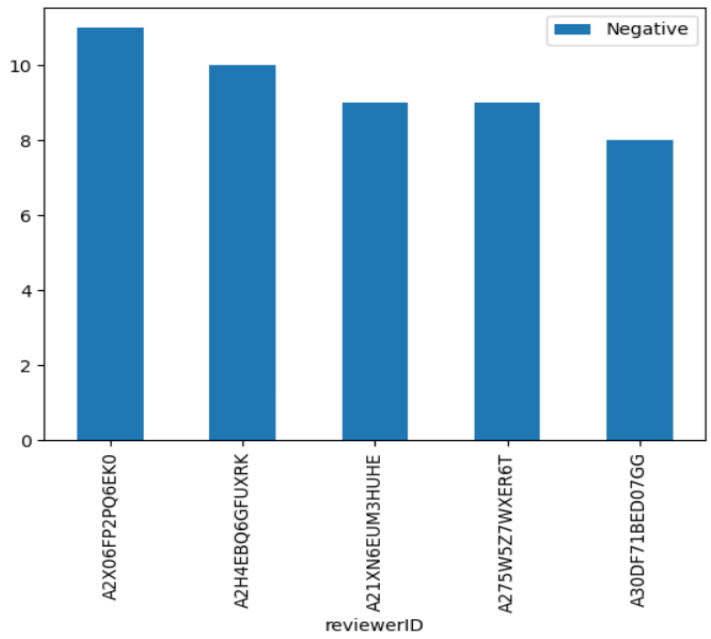
These customers are shopping in large numbers and giving positive reviews.

	reviewerID	Item_bought	Negative	Neutral	Positive	Amount	Verified_True	Verified_False
1826	A1M9D9VS6NFOA7	39	3	0	37	2391.37	39	1
4559	A2J2MQ45S3N25N	39	2	0	39	536.21	39	2
4635	A2JZUUAAK3HNHS	55	1	17	37	1098.06	55	0
5528	A2W19IAPS7ZSDK	55	1	3	57	992.56	55	6
7579	A3L1VJMHFWONCB	81	2	8	74	1163.24	81	3



Dissatisfied Customers:

	reviewerID	Item_bought	Negative	Neutral	Positive	Amount	Verified_True	Verified_False
5613	A2X06FP2PQ6EK0	8	11	1	9	489.23	8	13
4389	A2H4EBQ6GFUXRK	15	10	3	2	752.15	15	0
3142	A21XN6EUM3HUHE	16	9	6	2	231.88	16	1
3602	A275W5Z7WXER6T	11	9	1	1	169.28	11	0
5910	A30DF71BED07GG	16	8	2	6	1349.60	16	0



6.8 Demo of Search Recommendation:

```
You searched for : phone

 Search Recommendations are :
```

column Price

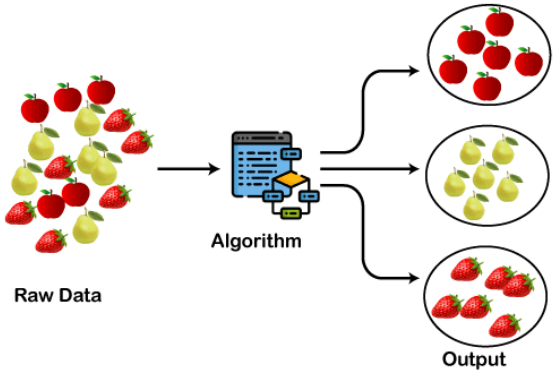
sort True

n 5

	asin	Total_Review	Rating	Price	brand
3	B0013P6ZJQ	7	4.750000	1.41	Technology Alternatives Corporation
9	B007POCIM2	6	5.000000	2.89	uxcell
24	B00MFRZ2SG	9	4.666667	3.29	uxcell
0	B00004Z2HR	10	5.000000	3.58	ARROW FASTENER
42	B01FHQ3V52	0	4.833333	3.92	General Tools

7. Clustering:

Clustering is the process of arranging a group of objects in such a manner that the objects in the same group (which is referred to as a cluster) are more similar to each other than to the objects in any other group.



In this project we clustered the customers those who are very loyal to particular product and those who are non-frequent buyer and also in similar manner we created cluster for different type of product on the basis of price what price range product is preferred by the customers in differentiate by Reviews.

7.1 Features Selection:

As described above we are working on both customers and products to therefore we have created two different dataframe for both customers and product.

Customer Data Frame:

	reviewerID	Total_Purchase	Negative	Neutral	Positive	Price	Sales	Rank	Verified_True	Verified_False	Rating
7579	A3L1VJMHFWONCB	81	2	8	74	13.848095	1163.24	82401.0	81	3	4.226190
4635	A2JZUUAAK3HNHS	55	1	17	37	19.964727	1098.06	68290.0	55	0	4.200000
5528	A2W19IAPS7ZSDK	55	1	3	57	16.271475	992.56	20692.0	55	6	4.672131
4559	A2J2MQ45S3N25N	39	2	0	39	13.078293	536.21	28446.0	39	2	4.682927
1826	A1M9D9VS6NFOA7	39	3	0	37	59.784250	2391.37	170394.0	39	1	4.700000

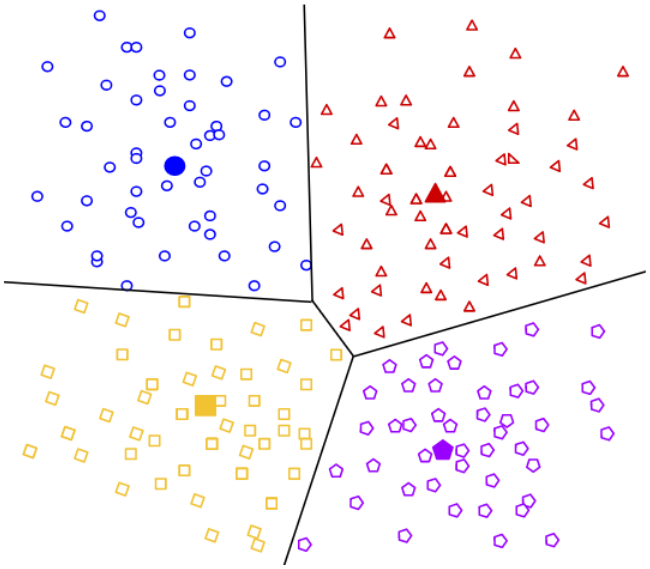
Product Data Frame:

	asin	Total_Sold	Negative	Neutral	Positive	Price	Sales	Rank	Verified_True	Verified_False	Rating
0	B0000223SI	12	0	0	13	15.99	207.87	19857.0	12	1	5
1	B0000223SK	20	0	1	21	17.99	395.78	13586.0	20	2	5
2	B0000223UV	26	1	2	32	10.24	358.40	340.0	26	9	5
3	B00002246J	15	0	1	15	4.34	69.44	330278.0	15	1	5
4	B0000224J0	12	0	0	15	6.98	104.70	51285.0	12	3	5

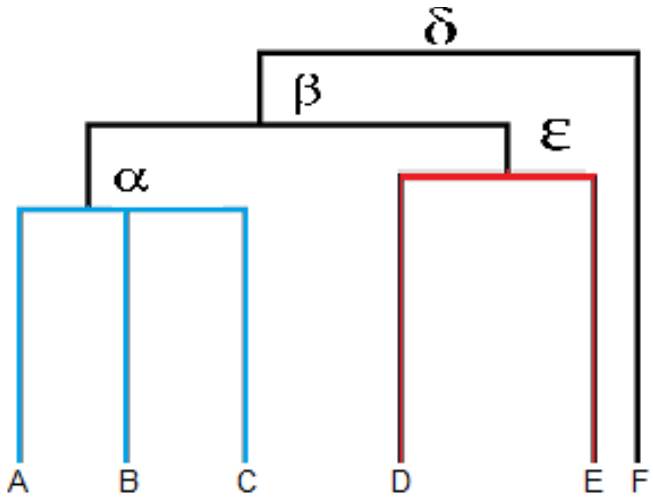
7.2 Type of Clustering Techniques:

There is multiple clustering method available for Each approach is best suited to a particular data distribution.
In our model we used two types of clustering first one is K Means basically a Centroid-based clustering and other is Agglomerative Clustering that is Hierarchical clustering.

7.2.1 Centroid-based clustering: The data point which is closest to the centroid of the cluster gets assigned to that cluster. After an iteration, it computes the centroids of those clusters again and the process continues until a pre-defined number of iterations are completed or when the centroids of the clusters do not change after an iteration.



7.2.2 Hierarchical clustering creates a tree of clusters. Hierarchical clustering, not surprisingly, is well suited to hierarchical data, such as taxonomies. In addition, another advantage is that any number of clusters can be chosen by cutting the tree at the right level.



7.3 Customers Clusters

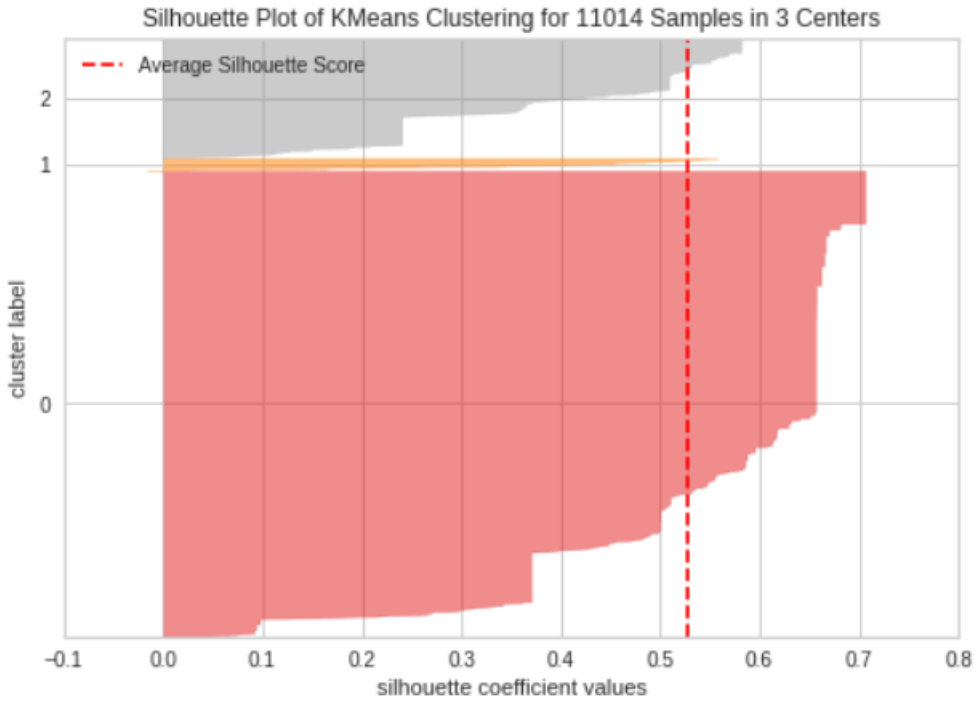
Here we clustered customers on the basis of their total purchase and reviews and ratings we used K-Mean clustering and Agglomerative Clustering in K Means clustering just to get the best value of K we use Silhouette Visualizer and dendrogram to find the number of clusters in Agglomerative Clustering.

	Total_Purchase	Negative	Neutral	Positive
0	3	0	0	5
1	5	0	0	5
2	8	0	2	7
3	7	0	1	6
4	8	0	2	7

7.3.1 K Means Clustering

Silhouette Visualizer: The Silhouette Coefficient is used when the ground-truth about the dataset is unknown and computes the density of clusters computed by the model. The score is computed by averaging the silhouette coefficient for each sample, computed as the difference between the average intra-cluster distance and the mean nearest-cluster distance for each sample, normalized by the maximum value. This produces a score between 1 and -1, where 1 is highly dense clusters and -1 is completely incorrect clustering.
The Silhouette Visualizer displays the silhouette coefficient for each sample on a per-cluster basis, visualizing which clusters are dense and which are not. This is particularly useful for determining cluster imbalance, or for selecting a value for K by comparing multiple visualizers.

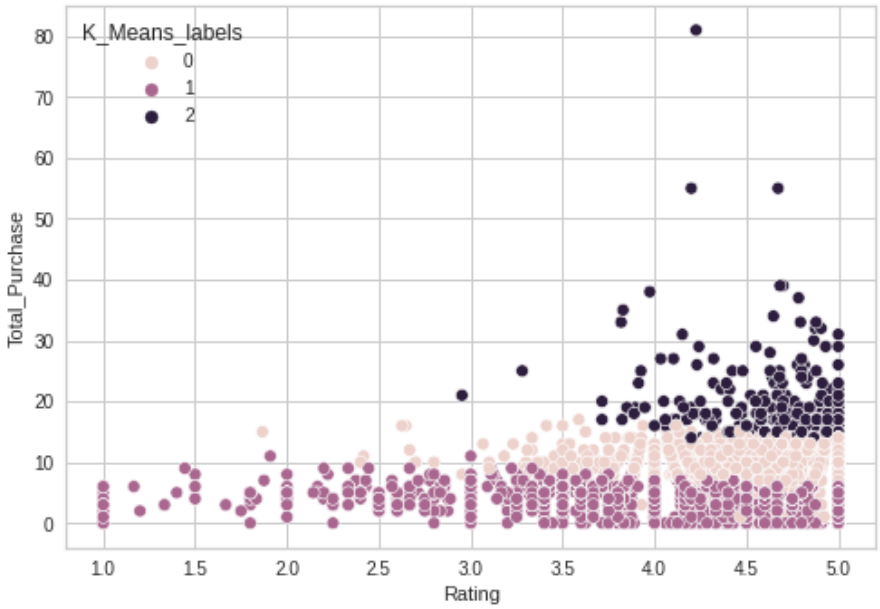
In the below graph we can see the with three cluster we are getting zero error and average Silhouette Score is high close to one.



Hence, we are now building our K-Means Model with K value 3.

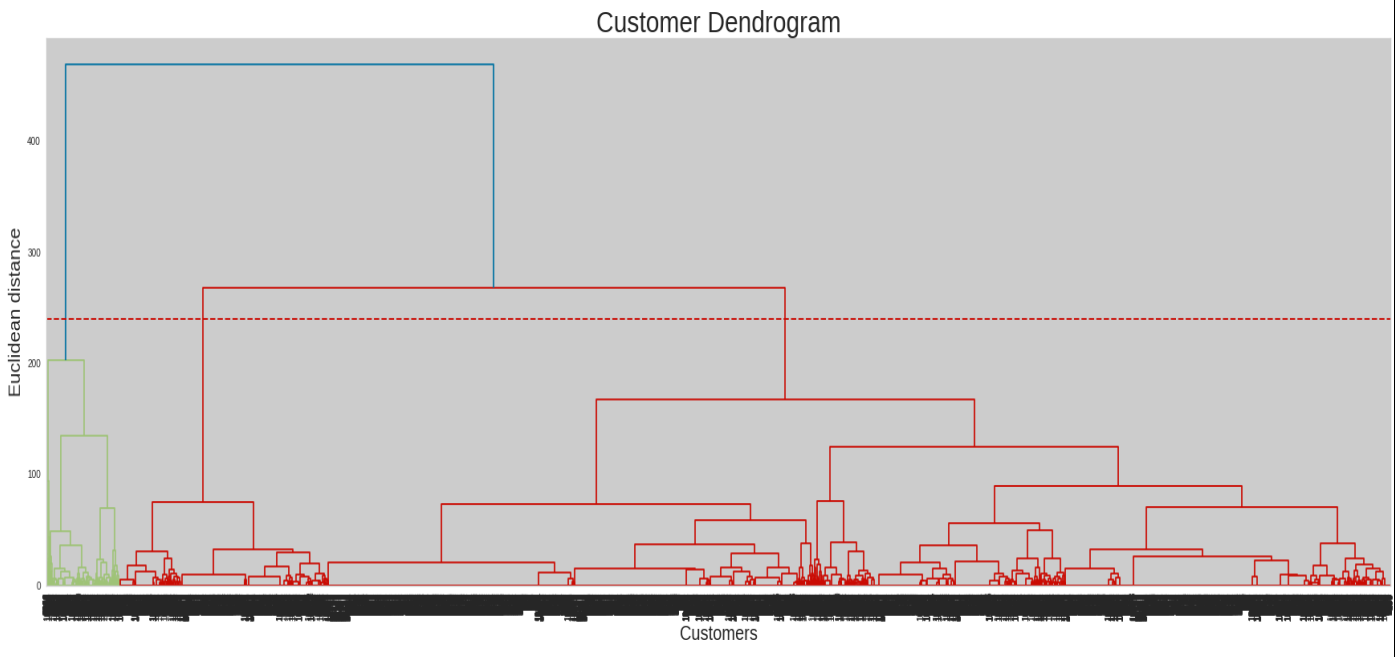
Result: With K-3 we got three different clusters which we defined as follow.

- **Loyal Customers**: 1st Cluster is for loyal customers where customers are more frequent purchasing hi gher number of pruducts and leaving a positive Review.
- **Opportunistic Buyers:** 2nd Cluster is for Opportunistic buyer an opportunistic buyer is likely to pay a much higher multiple when introduced to a company with overly attractive attributes, ones that are c oveted by the buyer.
- **Infrequent Buyers**: Cluster Zero is for Infrequent Buyers These customers are not frequent and purch ase very a smaller number of products their ratings can anywhere from one to five.



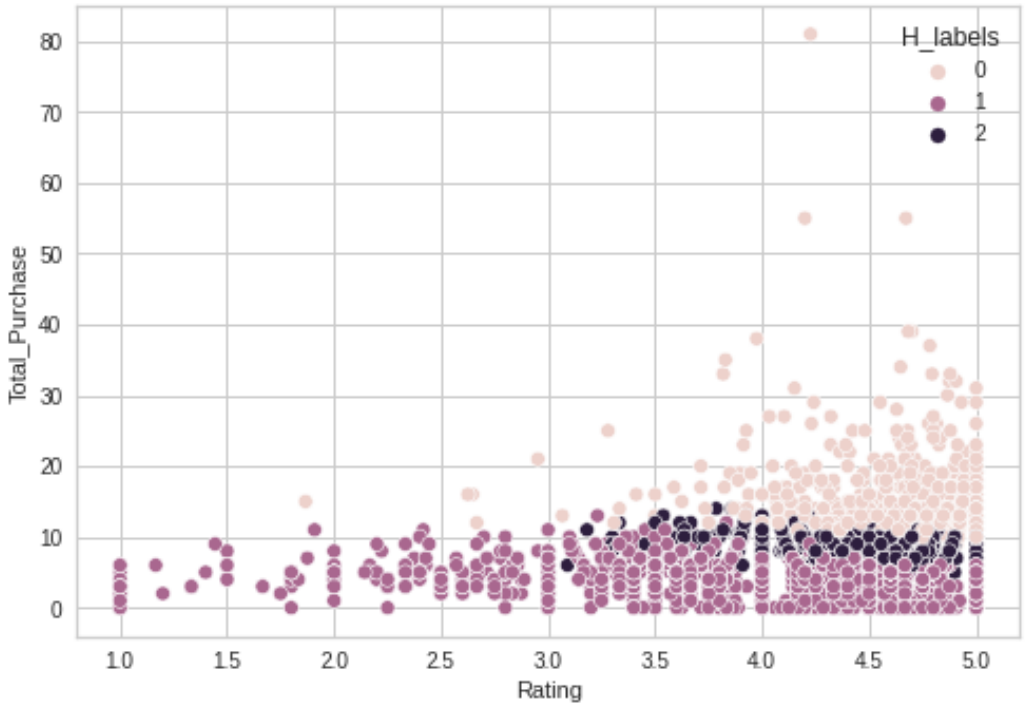
7.3.2 Agglomerative Clustering:

Dendrogram is a type of tree diagram showing hierarchical clustering — relationships between similar sets of data. They are frequently used in biology to show clustering between genes or samples, but they can represent any type of grouped data. The clades are arranged according to how similar (or dissimilar) they are. Clades that are close to the same height are similar to each other; clades with different heights are dissimilar — the greater the difference in height, the more dissimilarity



From the above Dendrogram we can see that there are three almost similar clades with same height. Hence, we are considering number of clusters n_clusters = 3

Result: We got the same result in Agglomerative Clustering but in second cluster it showing more customers in range of 10 Purchase product and for cluster zero we cannot define as loyal customers because in this model purchasing amount is not that high as K-Means because of these few reasons we are considering K-Means as our Final Model.



7.3.3 Final Result

We found three different clusters in our data and looking at the numbers we can say that loyal customers are very less most of the customers are opportunistic buyers and infrequent buyers but one insight that we got is most of our customers is satisfied with our product.

```
K_Means_labels
Infrequent Buyers       7179
Loyal Customers          382
Opportunistic Buyers    3453
Name: reviewerID, dtype: int64
```

```
H_labels
Infrequent Buyers       8704
Loyal Customers          603
Opportunistic Buyers    1707
Name: reviewerID, dtype: int64
```

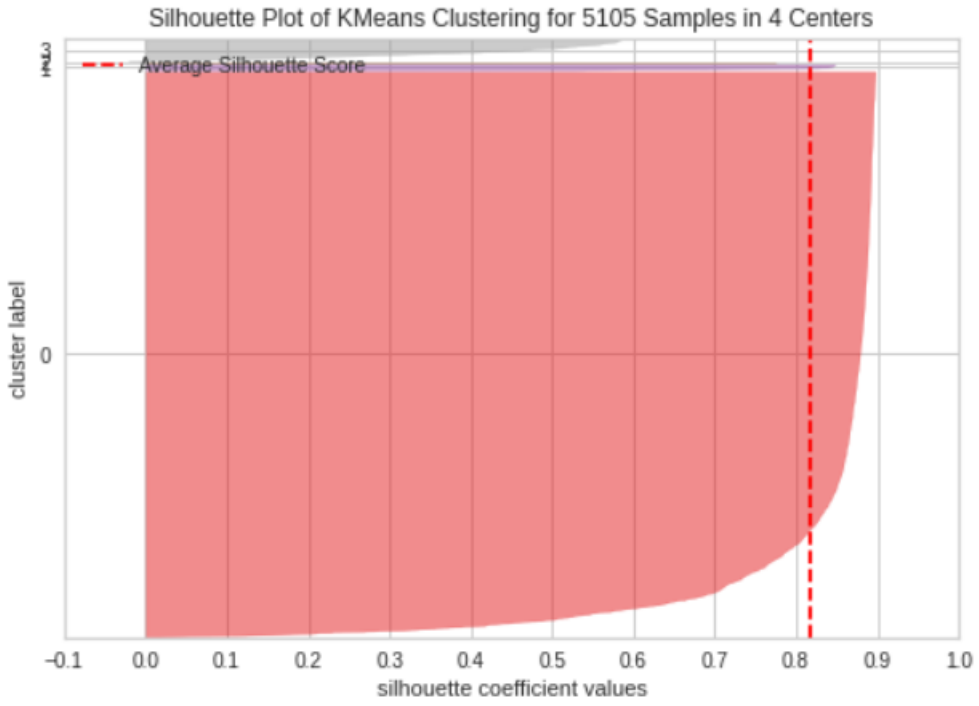
7.4 Product Clusters

Here we clustered product on the basis of number total sold products and price we used K-Mean clustering in K Means clustering just to get the best value of K we use Silhouette Visualizer.

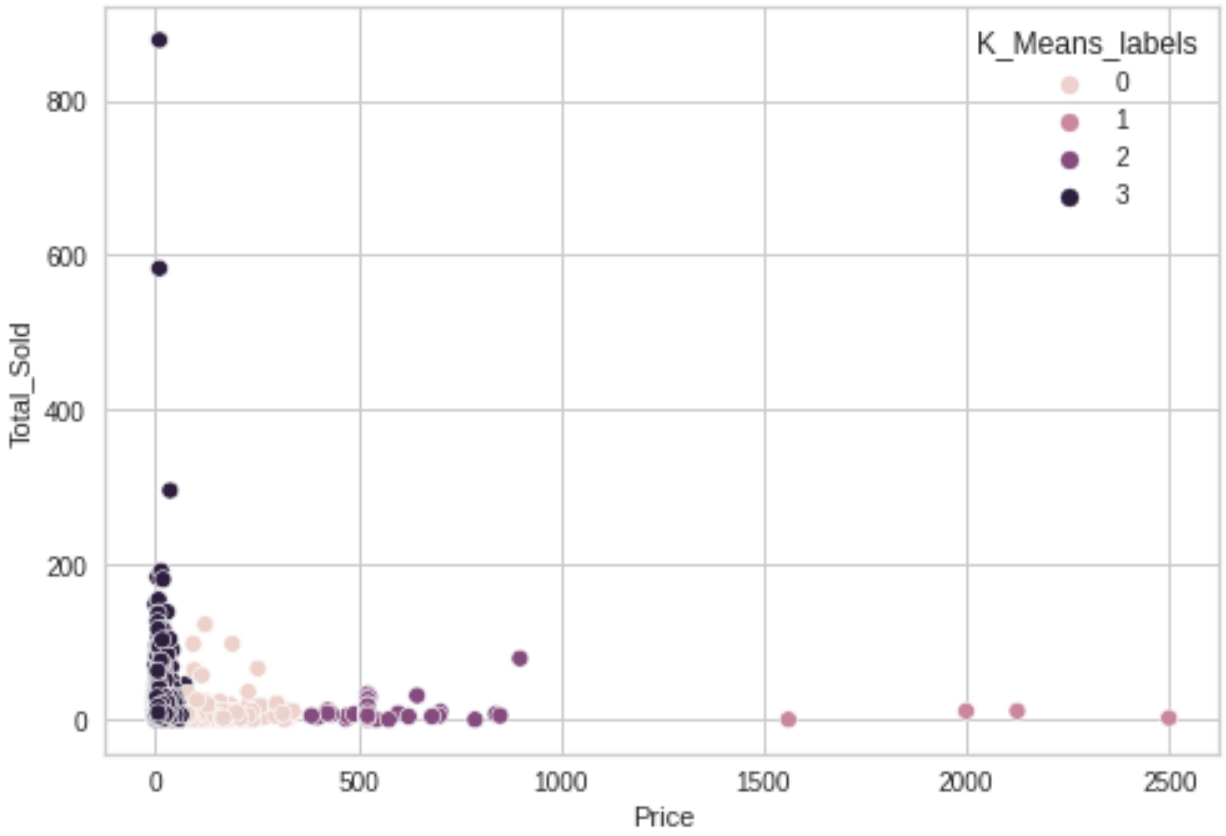
	Total_Sold	Price
0	12	15.99
1	20	17.99
2	26	10.24
3	15	4.34
4	12	6.98

7.4.1 K Means Clustering

In the below graph we can see the with three cluster we are getting zero error and average Silhouette Score is high close to one, Hence, we are now building our K-Means Model with K value 3.



7.4.2 Result: With K-4 we got four different clusters which we defined as follow.



- Low price high volume: Cluster 0 was identified as products that have low price and sell in large volume.
- High price low volume: Cluster 1 was identified as products that have high price and sell in low volume.
- Astronomical price very low volume: Cluster 2 was identified as products that are astronomically priced a sell in extremely low volume.
- Avg. price avg. volume: Cluster 3 was identified as products that have an average price and sell in average quantities.

```
K_Means_labels
Astronomical Price Very low Volume      53
Avg. Price Avg Volume                    4
High Price Low Volume                 4861
Low Price High Volume                  187
Name: asin, dtype: int64
```

8. Time Series:

Time Series Analysis is the way of studying the characteristics of the response variable with respect to time, as the independent variable. To estimate the target variable in the name of predicting or forecasting, use the time variable as the point of reference. In this article we will discuss in detail TSA Objectives, Assumptions, Components (stationary, and non-stationary). Along with the TSA algorithm and specific use cases in Python

Objectives:

- To understand how time series works, what factors are affecting a certain variable(s) at different points of time.
- Time series analysis will provide the consequences and insights of features of the given dataset that changes over time.
- Supporting to derive the predicting the future values of the time series variable.
- Assumptions: There is one and the only assumption that is "stationary", which means that the origin of time, does not affect the properties of the process under the statistical factor.

8.1 Components of Time Series Analysis

- **Trend**: In which there is no fixed interval and any divergence within the given dataset is a continuous timeline. The trend would be Negative or Positive or Null Trend
- **Seasonality**: In which regular or fixed interval shifts within the dataset in a continuous timeline. Would be bell curve or saw tooth
- **Cyclical**: In which there is no fixed interval, uncertainty in movement and its pattern
- **Irregularity**: Unexpected situations/events/scenarios and spikes in a short time span.

8.2 Data Types of Time Series

Let's discuss the time series' data types and their influence. While discussing TS data-types, there are two major types.

- Stationary
- Non- Stationary

8.2.1 Stationary: A dataset should follow the below thumb rules, without having Trend, Seasonality, Cyclical, and Irregularity component of time series

- The MEAN value of them should be completely constant in the data during the analysis
- The VARIANCE should be constant with respect to the time-frame
- The COVARIANCE measures the relationship between two variables.

8.2.2 Non- Stationary: This is just the opposite of Stationary.

8.3 Methods to check Stationarity

During the TSA model preparation workflow, we must access if the given dataset is Stationary or NOT. Using **Statistical and Plots test.**

8.3.1 Statistical Test:

There are two tests available to test if the dataset is Stationary or NOT.

- Augmented Dickey-Fuller (ADF) Test
- Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test

8.3.2 Augmented Dickey-Fuller (ADF)

Test or Unit Root Test: The ADF test is the most popular statistical test and with the following assumptions.

- Null Hypothesis (H0): Series is non-stationary
- Alternate Hypothesis (HA): Series is stationary
  - p-value >0.05 Fail to reject (H0)
  - p-value <= 0.05 Accept (H1)

8.4 Converting Non- stationary into stationary

Let's discuss quickly how to convert non-stationary into stationary for effective time series modelling. There are two major methods available for this conversion.

- Detrending
- Differencing

8.4.1 Differencing:

This is a simple transformation of the series into a new time series, which we use to remove the series dependence on time and stabilize the mean of the time series, so trend and seasonality are reduced during this transformation.

$Yt = Yt - Yt\text{-}1$

Yt=Value with time

8.4.2 Moving Average Methodology

The commonly used time series method is Moving Average. This method is slick with random short-term variations. Relatively associated with the components of time series.

The Moving Average (MA) (Or) Rolling Mean: In which MA has calculated by taking averaging data of the time-series, within k periods.

Let's see the types of moving averages:

- Simple Moving Average (SMA),
- Cumulative Moving Average (CMA)
- Exponential Moving Average (EMA)

8.5 Time Series Analysis in Data Science and Machine Learning

When dealing with TSA in Data Science and Machine Learning, there are multiple model options are available. In which the Autoregressive–Moving-Average (ARMA) models with [p, d, and q].

- P==> autoregressive lags
- q== moving average lags
- d==> difference in the order

Before we get to know about Arima, first you should understand the below terms better.

- Auto-Correlation Function (ACF)
- Partial Auto-Correlation Function (PACF)

8.5.1 Auto-Correlation Function (ACF):

ACF is used to indicate and how similar a value is within a given time series and the previous value. (OR) It measures the degree of the similarity between a given time series and the lagged version of that time series at different intervals that we observed.
Python Stats models library calculates autocorrelation. This is used to identify a set of trends in the given dataset and the influence of former observed values on the currently observed values.

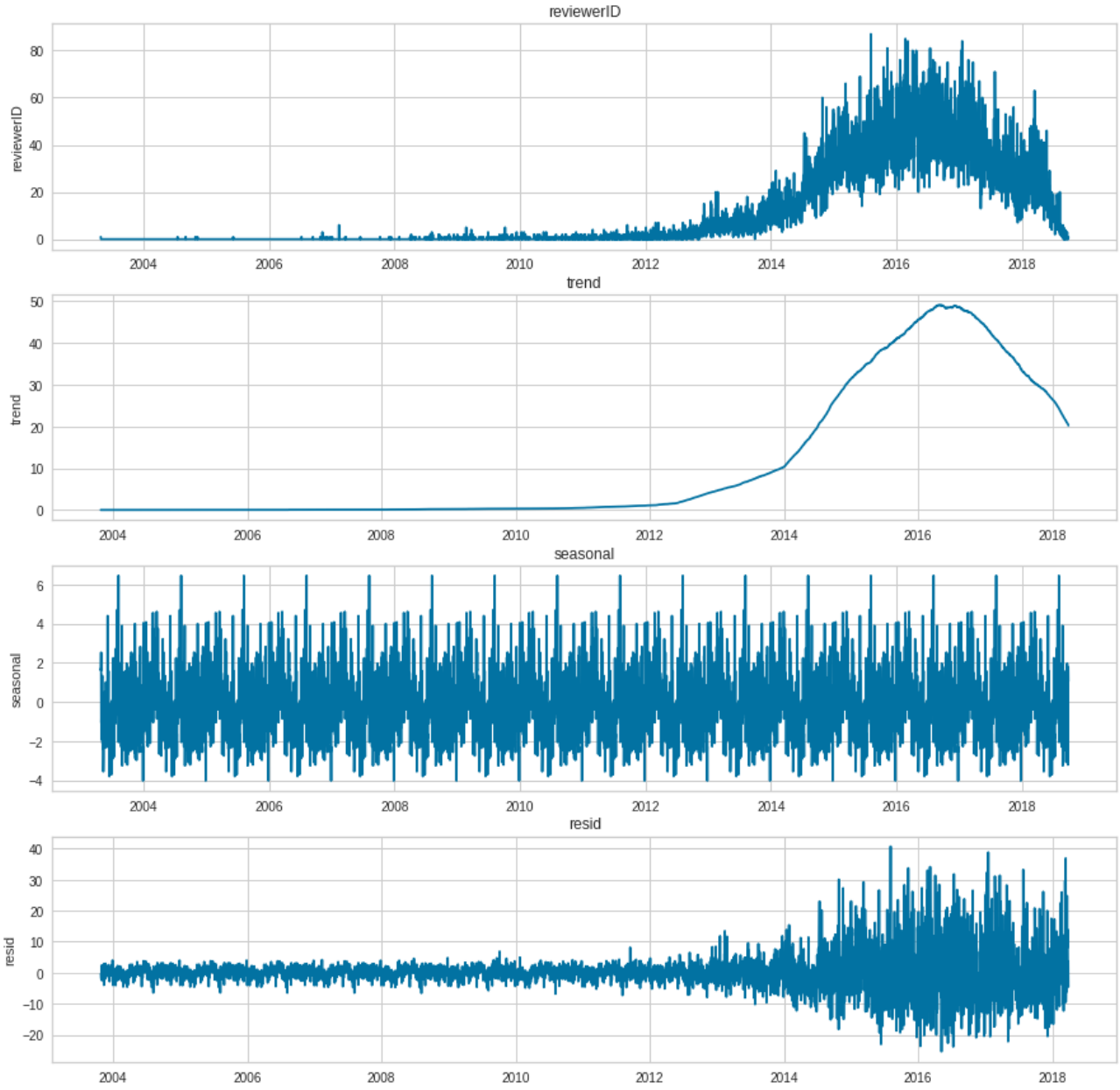
8.5.2 Partial Auto-Correlation (PACF):

PACF is similar to Auto-Correlation Function and is a little challenging to understand. It always shows the correlation of the sequence with itself with some number of time units per sequence order in which only the direct effect has been shown, and all other intermediary effects are removed from the given time series.

Time Series Analysis was intended to be performed on the number of reviews being written in a day.

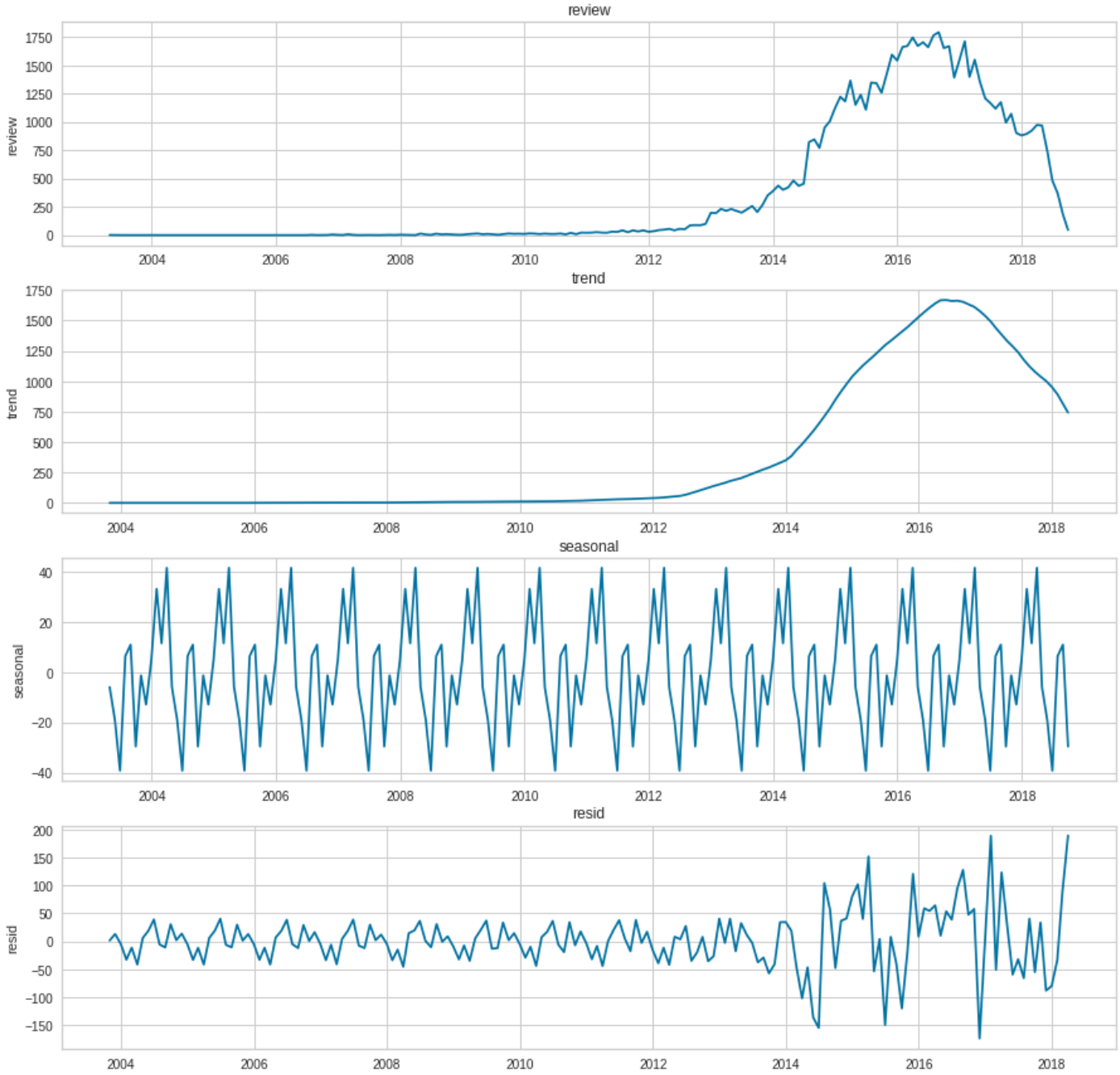
8.6 Data Decomposition:

Daily Data Decomposition



After the regrouped (day wise) data was decomposed into its trend, seasonality and irregular components it was observed that the data has a clear positive and a hazy seasonality component was observed in bad resolution. To be able to have a discernible seasonality component it was decided performing time series analysis on the data after regrouping by month will yield better results.

Monthly Resampled Data Decomposition:

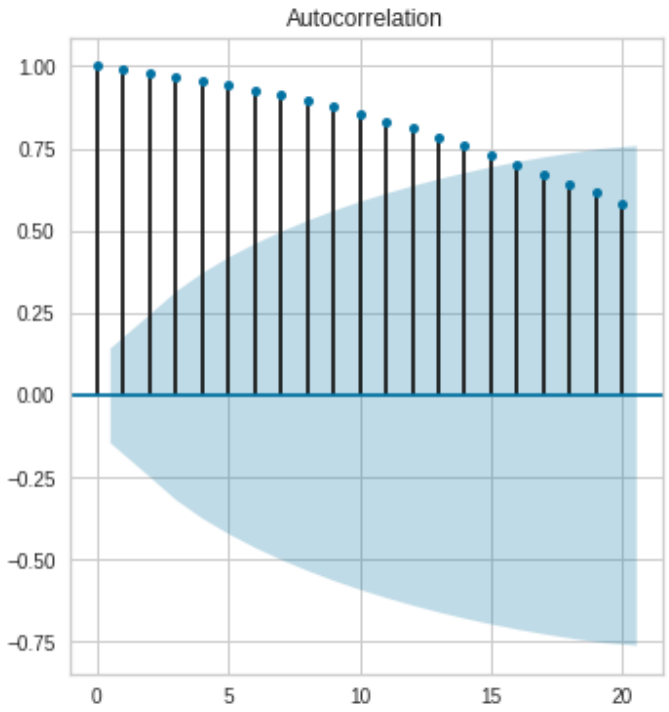


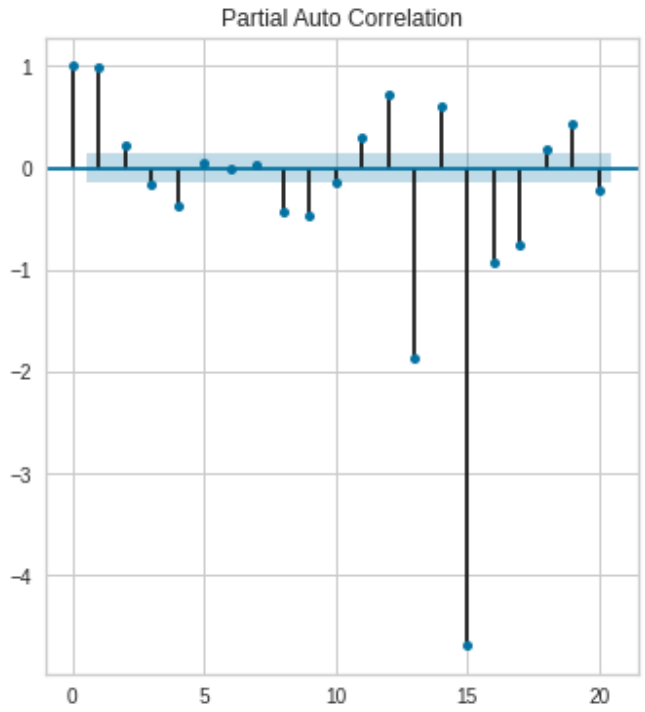
We made a function to view the autocorrelation, partial autocorrelation and stationarity of the data of a particular category or sentiment so that we could later perform time series to predict whether the number of positive reviews of a particular category go down or up.

8.7 SARIMAX

SARIMAX is a time series forecasting model that combines Seasonal Autoregressive Integrated Moving Average (SARIMA) with an exogenous variable. SARIMA is a type of time series model that models the relationship between the current value of a time series and its past values, as well as its past seasonal values. The exogenous variable is a variable from outside the time series that can help improve the accuracy of the forecast. SARIMAX is useful for forecasting time series data with a strong seasonal component, such as sales or energy demand data, where the values of the time series are influenced by the season of the year. The parameters of SARIMAX, such as the order of differencing, the order of autoregression, the order of moving average, and the order of seasonality, can be estimated using statistical techniques to obtain the best forecast.

8.7.1 Overall Reviews:



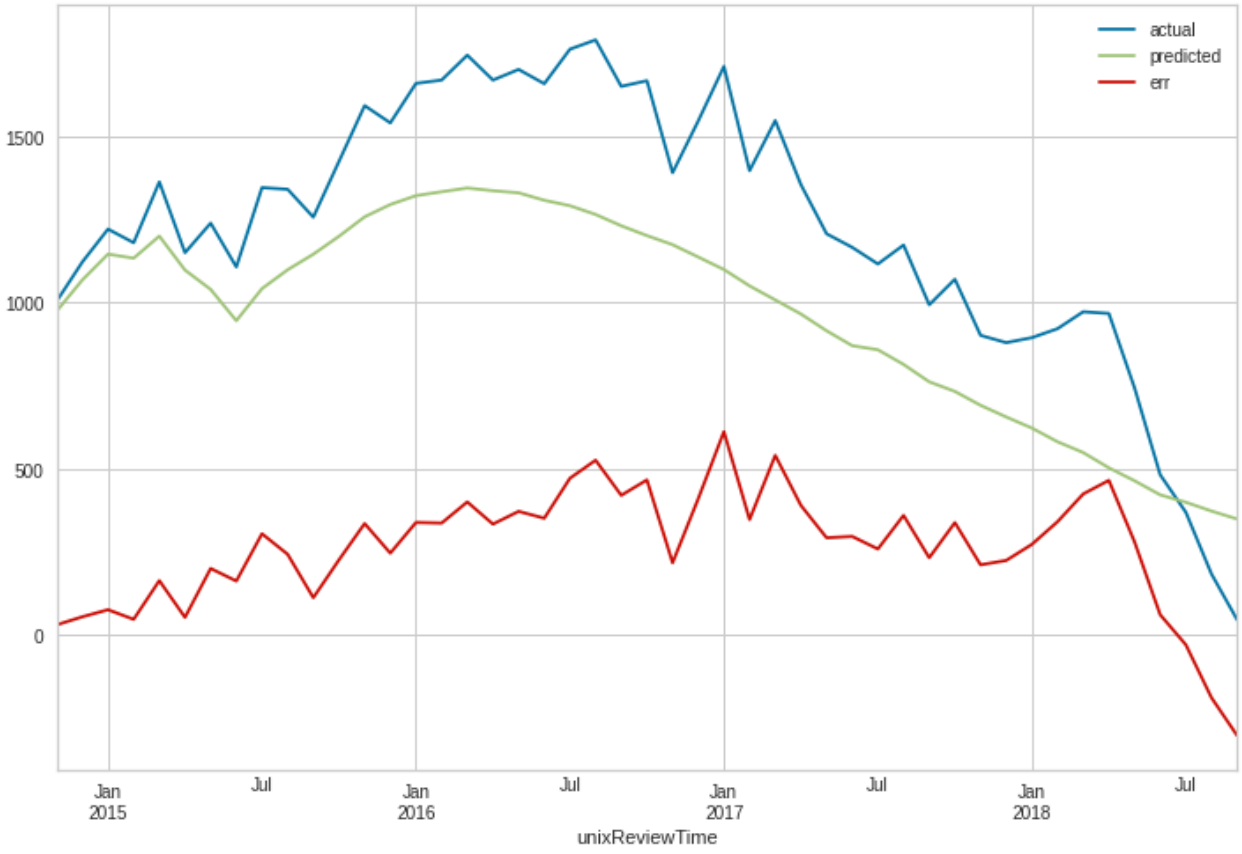


After verifying that the data was stationary and taking p and q values from ACF and PACF plots we chose to apply SARIMAX model first on the overall data and then on sections to predict the number of reviews on a particular category with respect to a particular sentiment.

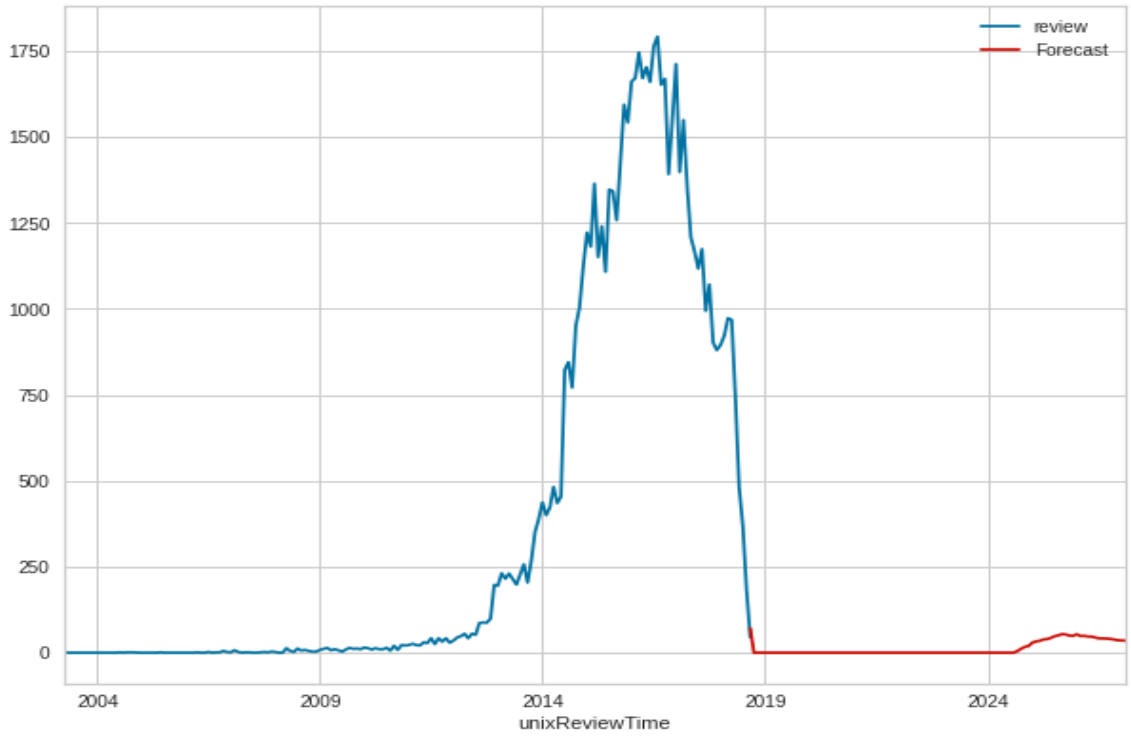
The summary of the model that was applied is seen below.

```
                                SARIMAX Results
==========================================================================================
Dep. Variable:                            review   No. Observations:                  186
Model:             SARIMAX (3, 0, 14) x (3, 0, 14, 18)   Log Likelihood               -
1035.441
Date:                           Fri, 20 Jan 2023   AIC                           2140.883
Time:                                   11:03:37   BIC                           2253.784
Sample:                             04-30-2003   HQIC                          2186.635
                                  - 09-30-2018
Covariance Type:                            opg
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):              1043.91
Prob(Q):                              0.97   Prob(JB):                         0.00
Heteroskedasticity (H):            1886.30   Skew:                             0.50
Prob(H) (two-sided):                  0.00   Kurtosis:                        14.56
===================================================================================
```

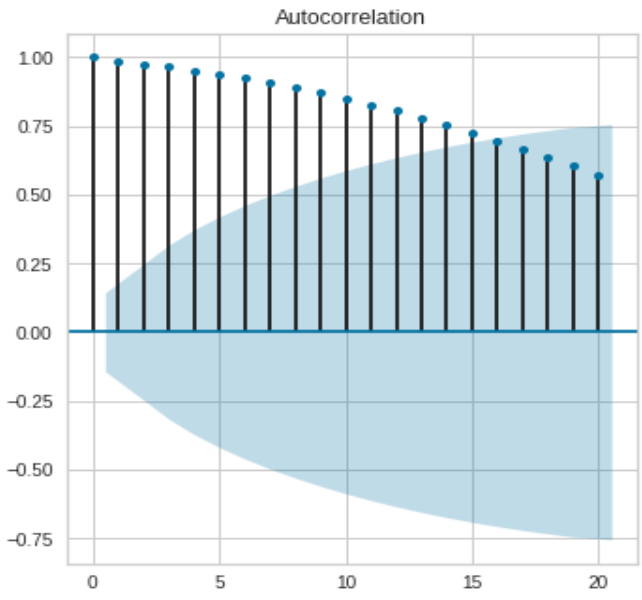
After applying the above model and comparing the error against our test data the error was as follows:

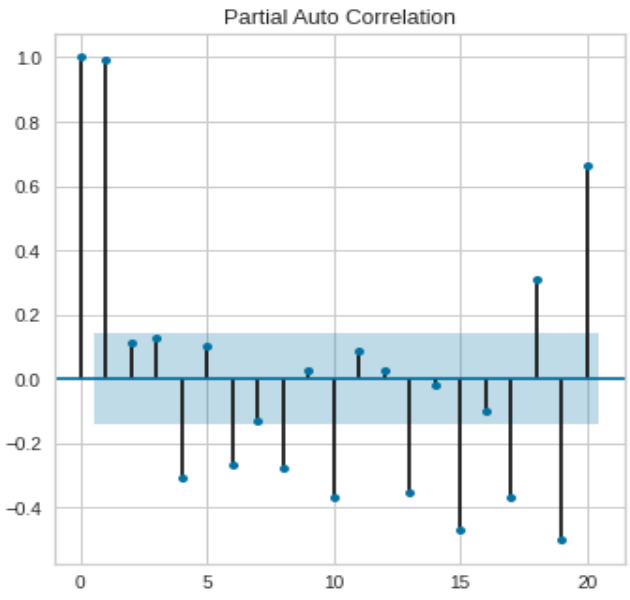


After receiving a reasonable error rate, we predicted the number of reviews for the near future using our past data that was fed to the model.



8.7.2 Positive Reviews:

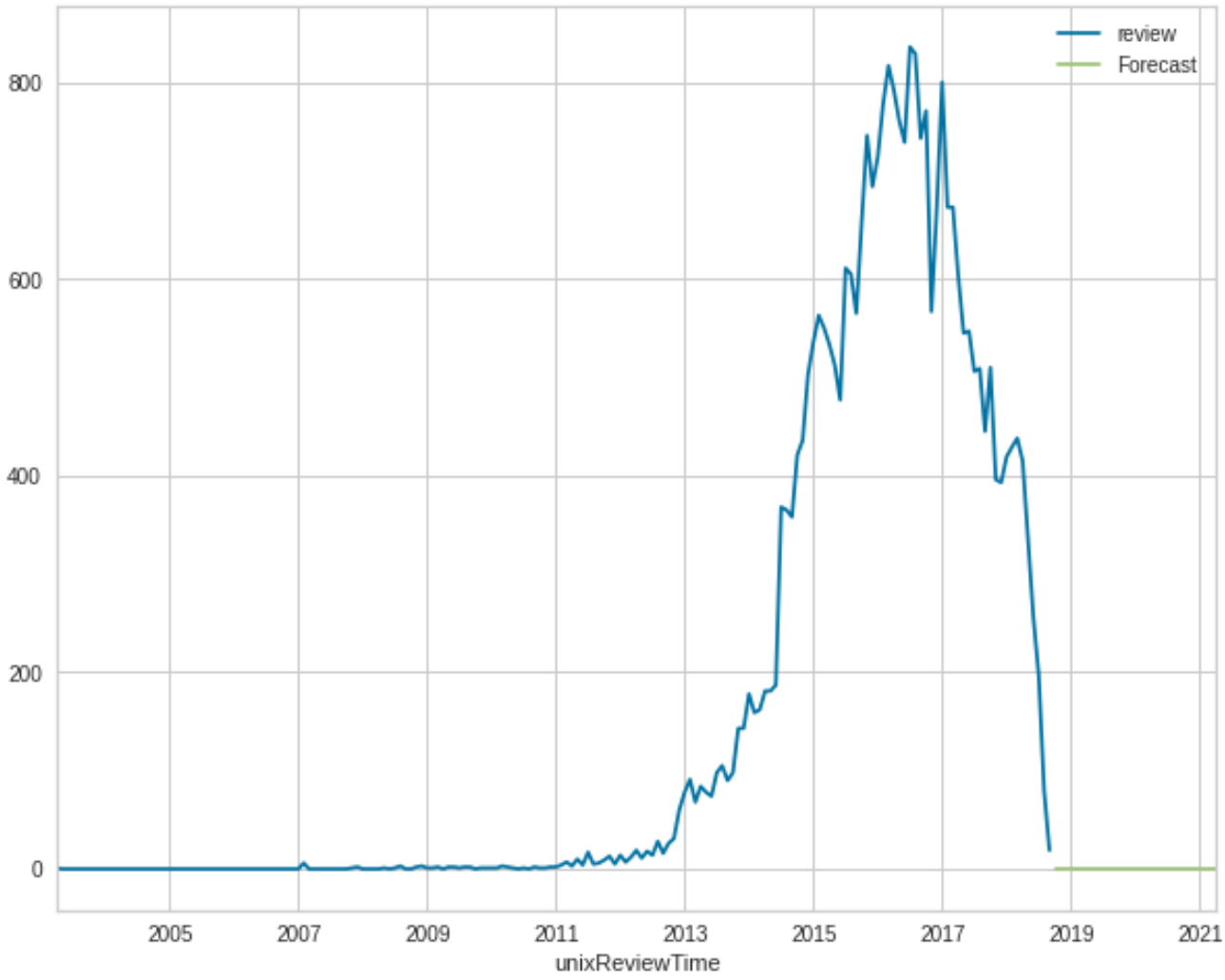




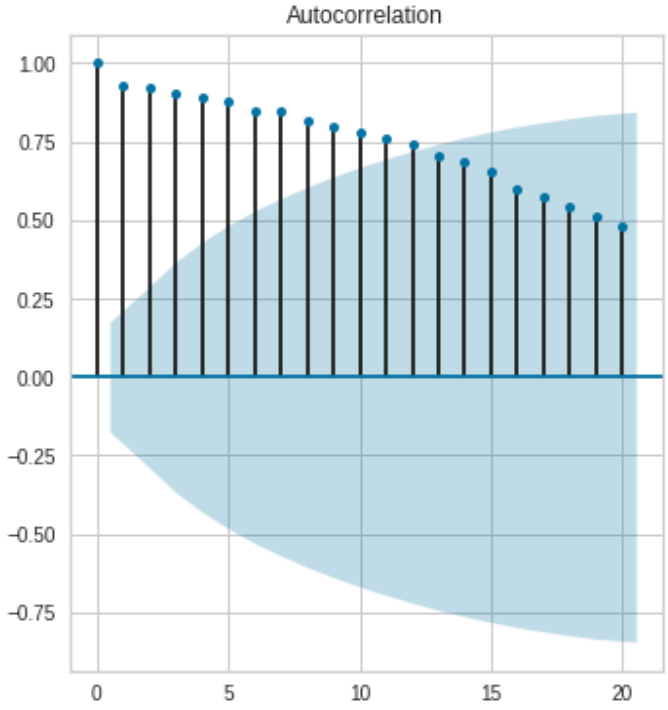
SARIMAX Results

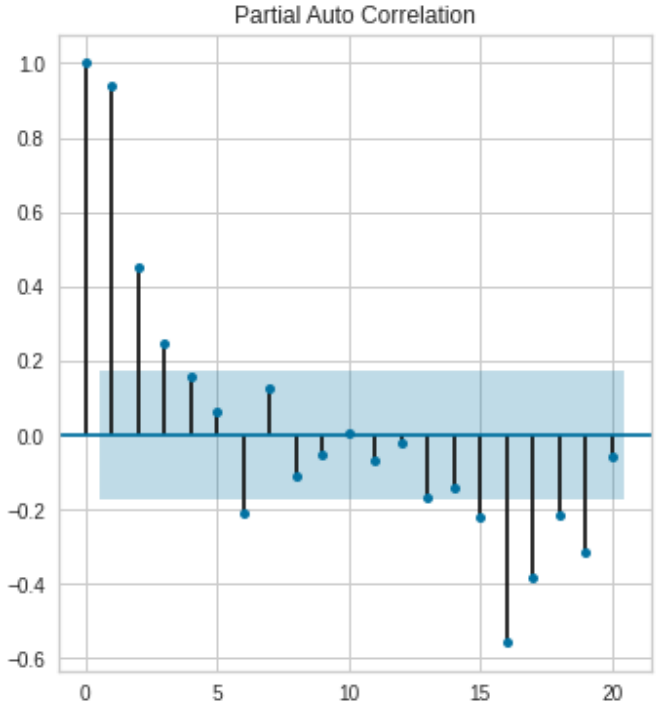
Dep. Variable:	review	No. Observations:	186
Model:	SARIMAX(3, 0, 12)x(3, 0, 12, 18)	Log Likelihood	-911.649
Date:	Fri, 20 Jan 2023	AIC	1885.298
Time:	11:57:11	BIC	1985.296
Sample:	04-30-2003 - 09-30-2018	HQIC	1925.821
Covariance Type:	opg		

Ljung-Box (L1) (Q):	0.05	Jarque-Bera (JB):	2367.89
Prob(Q):	0.82	Prob(JB):	0.00
Heteroskedasticity (H):	2165.07	Skew:	0.35
Prob(H) (two-sided):	0.00	Kurtosis:	20.47



8.7.3 Negative Reviews :

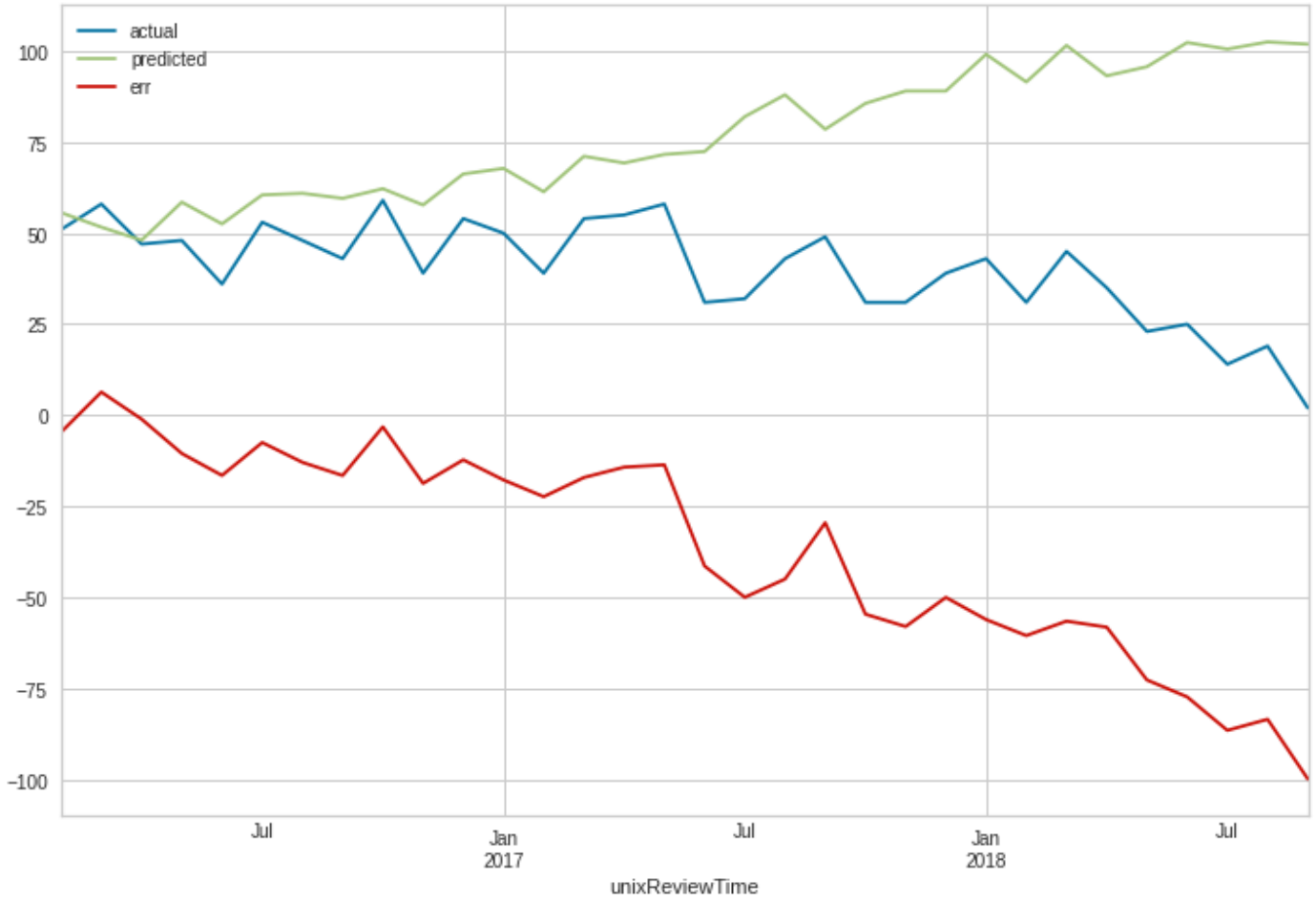


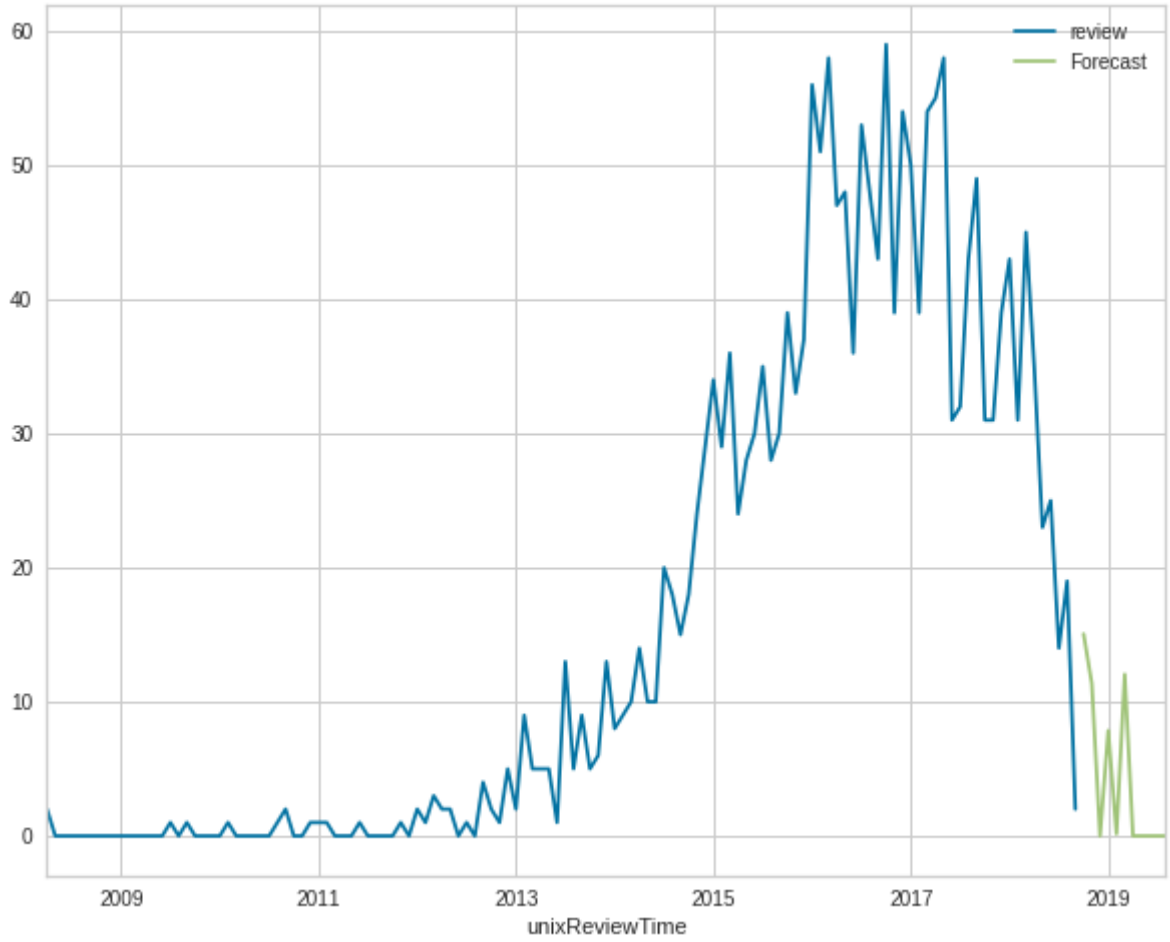


SARIMAX Results

Dep. Variable:	review	No. Observations:	126
Model:	SARIMAX(3, 1, 10)x(3, 1, 10, 18)	Log Likelihood	-338.559
Date:	Fri, 20 Jan 2023	AIC	731.118
Time:	12:05:07	BIC	803.285
Sample:	04-30-2008 - 09-30-2018	HQIC	760.373
Covariance Type:	opg		

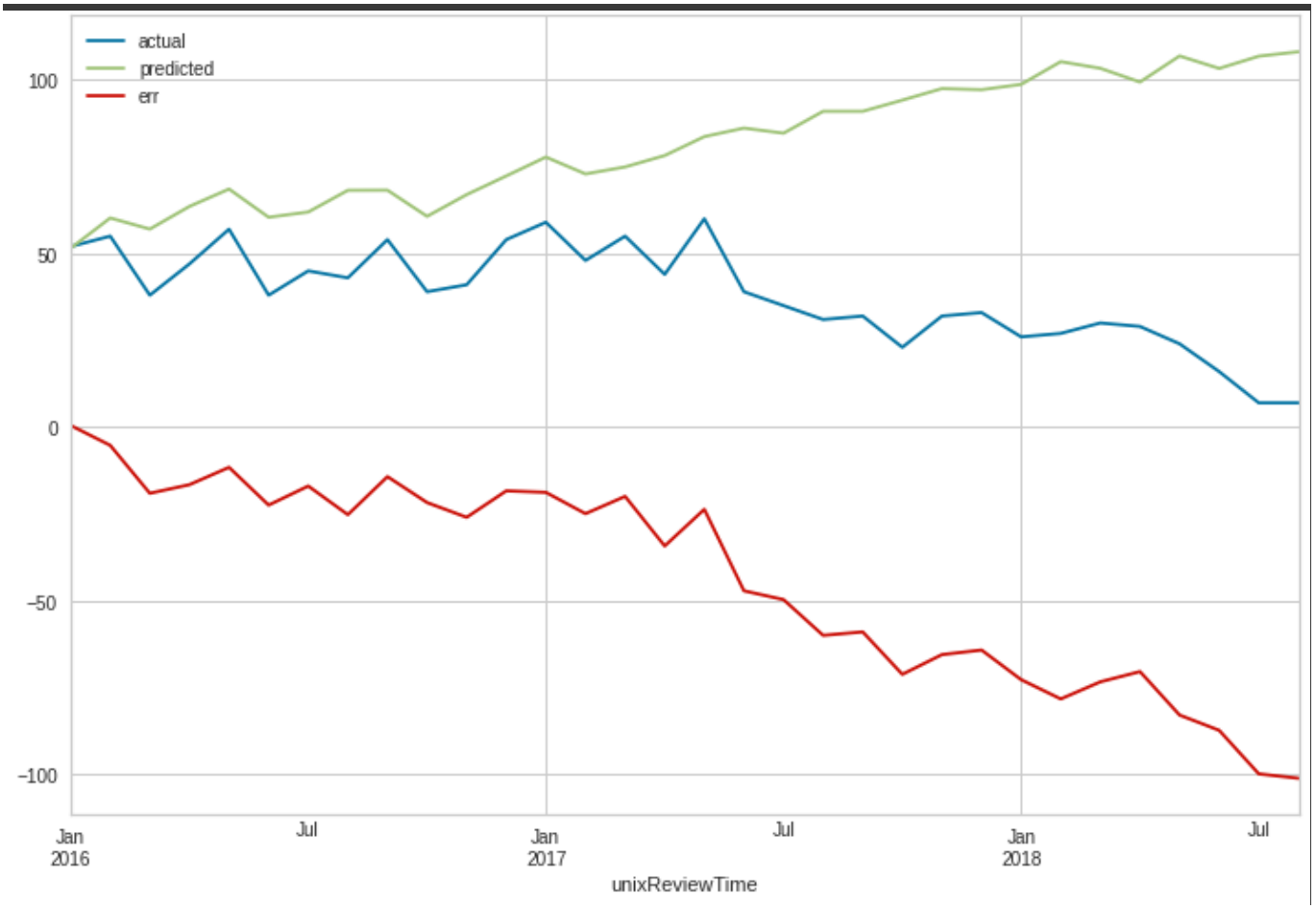
Ljung-Box (L1) (Q):	0.02	Jarque-Bera (JB):	31.74
Prob(Q):	0.90	Prob(JB):	0.00
Heteroskedasticity (H):	57.99	Skew:	-0.64
Prob(H) (two-sided):	0.00	Kurtosis:	5.34

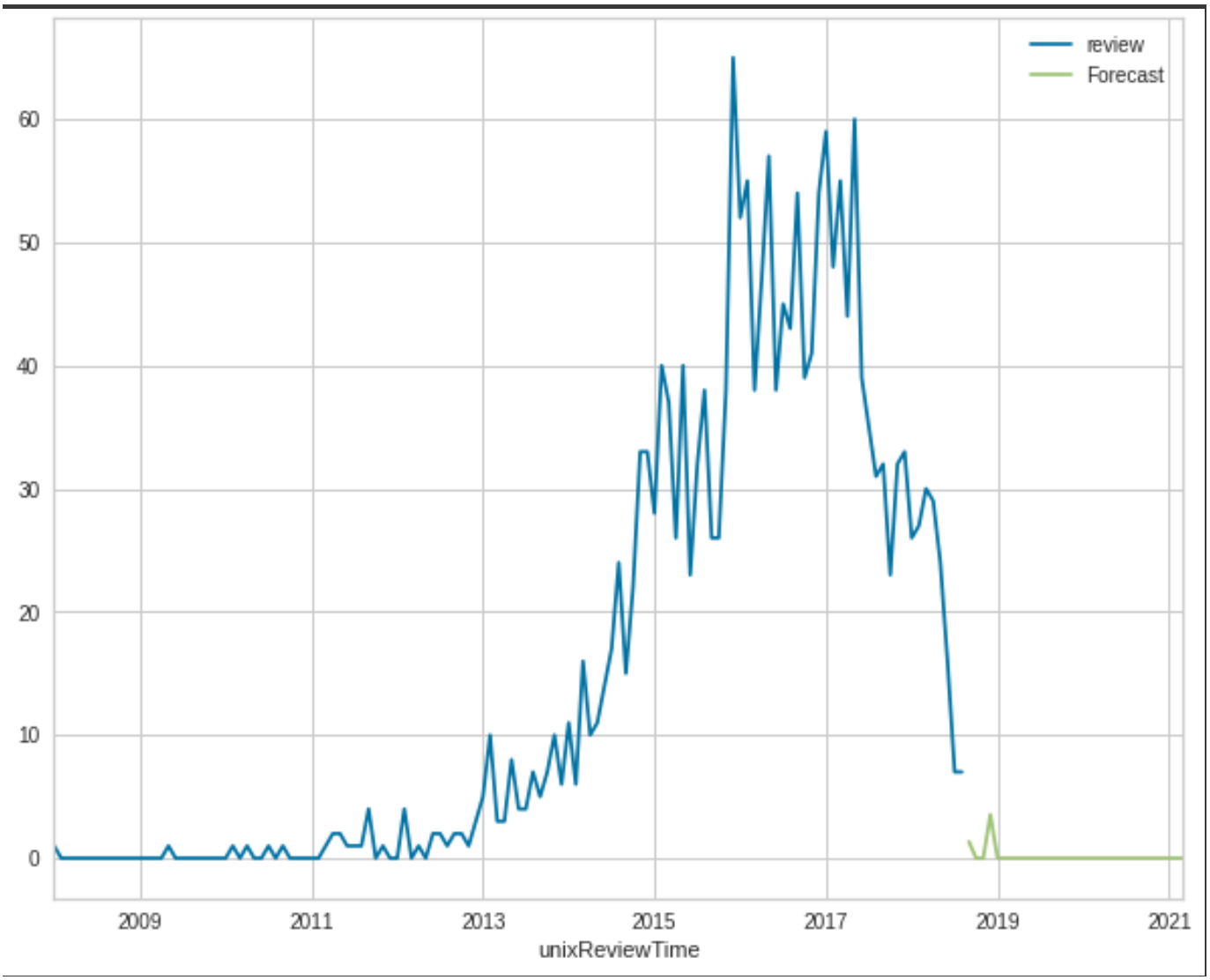




8.7.4.Neutral Reviews :

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                            review   No. Observations:                  128
Model:             SARIMAX(4, 1, 11)x(4, 1, 11, 18)   Log Likelihood                -
352.077
Date:                              Fri, 20 Jan 2023   AIC                            766.154
Time:                                      12:16:58   BIC                            849.585
Sample:                                  01-31-2008   HQIC                           799.988
                                       - 08-31-2018
Covariance Type:                               opg
==========================================================================================

Ljung-Box (L1) (Q):                   0.05   Jarque-Bera (JB):                35.59
Prob(Q):                              0.81   Prob(JB):                         0.00
Heteroskedasticity (H):              53.25   Skew:                             0.06
Prob(H) (two-sided):                  0.00   Kurtosis:                         5.80
==========================================================================================
```


actual
predicted
err
100
50
0
−50
−100
Jan
2016
Jul
Jan
2017
Jul
Jan
2018
Jul
unixReviewTime


review
Forecast
60
50
40
30
20
10
0
2009
2011
2013
2015
2017
2019
2021
unixReviewTime

9. REFERENCES

1. https://www.google.co.in/
2. https://scikit-learn.org/stable/
3. https://bootcamp.analyticsvidhya.com/?utm_source=google&utm_medium=AnalyticsVidhya&utm_campaign=Bootcamp_New_Search_BrandAV_EM_Geny&utm_content=search_console_phrase&utm_term=analytics%20vidhya&gclid=CjwKCAiAuOieBhAIEiwAgjCvcg6OBOGIlsPCInhBPsjkCwbP22Hdt1i7XJwy5CB6-vYC0hH9c-LWJxoCM5UQAvD_BwE
4. https://www.javatpoint.com/
5. https://www.google.com/imghp?hl=en
6. https://www.amazon.in/?&ext_vrnc=hi&tag=googhydrabk1-21&ref=pd_sl_1jyasdi57f_e&adgrpid=60456322738&hvpone=&hvptwo=&hvadid=486393568006&hvpos=&hvnetw=g&hvrand=6177243328587057658&hvqmt=e&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=1007765&hvtargid=kwd-295905178780&hydadcr=14451_2154369&gclid=CjwKCAiAuOieBhAIEiwAgjCvcnLYuYxcTW51DSbQjAPDyEh3haZC0ip9hOmpiBt8jlG5JCxdZv-enxoCp9UQAvD_BwE
7. https://jmcauley.ucsd.edu/data/amazon/