
VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

## SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

***J Component report***

## Programme: MCA

## Data Mining and Business Intelligence (ITA5007)

## *Slot*: A2+TA2

# AMAZON PRODUCT REVIEWS USING SENTIMENT ANALYSIS

‘

*Submitted in partial fulfillment for the course*

*by*

**NIKHIL SARAOGI (21MCA1080)**

June, 2022

**FACULTY NAME** **FACULTY SIGN**

**DR PUNITHA K**


## DECLARATION

I hereby declare that the project entitled "**AMAZON PRODUCT REVIEWS USING SENTIMENT ANALYSIS**" submitted by me; for the completion of the course, Data mining and Business Intelligence (ITA5007) is a record of bonafide work carried out by me under the supervision of Dr Punitha K, my course instructor. I further declare that the work reported in this document has not been submitted and will not be submitted, either in part or in full, for any other courses in this institute or any other institute or university.

Place: Chennai

Date: 02-06-2022 Signature of the Candidate


## School of Computer Science and Engineering

# CERTIFICATE

This is to certify that the report entitled **"AMAZON PRODUCT REVIEWS USING SENTIMENT ANALYSIS"** is prepared and submitted by **NIKHIL SARAOGI** (**21MCA1080**) to Vellore Institute of Technology, Chennai, in partial full fillment of the requirement for the course, **Data Mining and Business Intelligence (ITA5007),** is a bonafide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the any other course and the same is certified.

Name: Dr.Punitha K Signature of the Faculty

Date:

# ABSTRACT

The intense competition to attract and maintain customers online is compelling businesses to implement novel strategies to enhance the customer experiences. It is becoming necessary for companies to examine customer reviews on online platforms such as Amazon to understand better how customers rate their products and services. Online stores like Amazon provide a website for consumers to express their opinions about different items. Since then, it has been established that buying online, 90% of consumers are testing different websites channels to determine the quality of their purchase. The purpose of this study is to investigate how companies can conduct sentiment analysis based on Amazon reviews to gain more insights into customer experiences. The dataset selected for this capstone consists of customer reviews and ratings from consumer reviews of Amazon products. Amazon product reviews enable a business to gain insights on customer experiences regarding specific products and services. The study will enable companies to pinpoint the reasons for positive and negative customer reviews and implement effective strategies to address them accordingly. The project helps companies use sentiment analysis to understand customer experiences using Amazon reviews.

## *Table of Contents*

*List of Figures*

**Chapter-1**

# 1.1. Background

This chapter entails the projects goals, aims and objectives, research methodology and limitations of the study. Amazon is among the largest online marketplace in the world for various products. Customers occasionally go through the products and their reviews just before they purchase a product. These reviews give them a level of information and opinion about the quality of the products they want to buy. Sometimes the reviews prove to be misleading because they are usually subjective and could not be giving the entire information about a product but just a first impression of a customer with the Amazon product or service. Data from customer reviews is critical in today's information-driven business environment. Companies use customer reviews to gain meaningful insight into the consumption behaviors of customers. An in-depth analysis of customer sentiments enables a business to understand the market better and make rational decisions to address customer needs and concerns proactively. Sentiment analysis capitalizes on natural processing language statistics and text analysis to explore what the customers are saying, how they express it, and what they mean. Tweets, reviews, and comments are crucial sources of customer sentiments. The computations of sentiment analysis determine whether customer reviews are positive, negative, neutral, or mixed. According to Du et al. (2019), 91% of online shoppers read product reviews before purchasing products and services online. Product reviews play a crucial role in enhancing the customer purchasing experience. Besides, it is essential to enable a business to improve its products and services by comprehending customers' needs, preferences, and tastes. However, it is imperative to note that product reviews on Amazon platforms are vulnerable to quality control. According to Du et al. (2019), a recent study revealed that customers tend to restrict concentration on the first view reviews, irrespective of their helpfulness. The challenge is that access to extensive customer reviews makes it difficult for customers to identify useful information.

## 1.2. Problem Statement

The retail industry is the backbone of the US economy. According to Statista projections, the sales revenue forecast for this industry in 2020 was $5.48 trillion (Statista.com, 2020). The retail sector creates millions of jobs annually, generates revenue, and contributes approximately 10% of the gross national product (Lim et al., 2019). The retail industry's health dramatically relies on the degree of customer satisfaction and confidence with the retailers of their preferred products and services. Advancements in technology enable customers to share their experiences by reviewing the products and services from specific retailers. Today, social media platforms and online networks allow businesses to mine genuine comments and reviews from customers worldwide. Customer reviews reveal customers' experiences regarding the prices, value, quality, customer service, ease of shopping, and more factors about what they shop online. The customer reviews are unstructured, and the sentiment analysis will help extract the sense of these unstructured texts efficiently and cost-effectively. This capstone project will study how companies can conduct sentiment analysis based on Amazon reviews to gain customer experience insights. Companies will gain more understanding of top-rated products and services, what customers value, and what they dislike using sentiment analysis. For a business to succeed in today's competitive and information-driven business environment, it is vital to understand what the customers feel about the products and services offered. A company must maintain positive reviews from customers and improve the neutral and negative reviews of the customers. For instance, most customers complain about the quality of products based on Amazon reviews, and the business must develop and implement effective strategies to enhance the quality of their offerings.

## 1.3. Project Goal

The first goal is to get the sentiments expressed in the customer reviews and analyze the frequency of the sentiments. The second project goal is to build and train a machine learning model that can be used to classify customer reviews into two sentiments (positive or negative).

## 1.4. Aims and Objectives

The aim of the study is to classify customer reviews into positive or negative sentiment.The objectives are as follows:

- To measure the intensity of the sentiments generated from the customer reviews
- To analyze the association between customer reviews concerning different Amazon products.

## 1.5. Research Methodology

Sentiment analysis methodology will be used in this research. The method is also referred to as opinion mining. The process relies on machine learning (ML) algorithms and natural language processing (NLP) to determine the emotions behind online reviews. The research will focus on analyzing the sentiments on Amazon product reviews. To be specific, the platform has a feature where customers can review products on a five-scale rating. The ratings from 5 to 1 represent very positive, positive, neutral, negative, and very negative experiences. Five stars mean the customer is very positive about the product or service. One star indicates that the customer's experience is very negative.

Sentiment analysis follows the certain steps to collect, process, transform and convert raw data to simple corpus for sentiment classification. These steps are essential in achieving the best results in sentiment analysis.

## STEPS :-

**STEP 1**: A data set of Amazon product reviews within a specific period will be collected. Once the data set is collected, the next step is text preparation. This is the process in which the data extracted is filtered before analysis.

**STEP 2:** In this stage, non-textual content is identified and eliminated. Irrelevant content is also identified and removed from the data set. The objective is to ensure that only the required data setis analyzed.

**STEP 3**: Sentiment detection. The purpose of sentiment detection is to review each comment for subjectivity. Sentences or comments with objective expressions are eliminated, while those with subjective expressions are retained further.

**STEP 4**: Sentiment classification. This stage's data is categorized into two major groups, positive and negative. The data can be classified more to include like and dislike categories.

**STEP 5**: The final step is to convert the results of the analysis into meaningful information. The text results will be displayed on bar charts, pie charts, and line charts. The graphs will visualize the results for a better understanding of the trends in online product reviews.

The data tools used in this project are: R Programming Languages, Amazon APIs, and Visualization tools (Tableau) will be used in this research.

# 1.6 Limitation of the Study

- The study was limited when it comes to dealing with reviews from sarcastic customers who usually use ironic language, this would make it hard for the model to learn the correct sentiment elicited.
- Another limitation arose in the subjectivity of the customers, this would hinder correct sentiment extraction from the reviews because the subjectiveness changes from person to person, and some people would be very irrational when submitting their reviews.
- People’s opinions change over time and this could be due to mood change or even interaction with other products and customers, hence when collecting data the period is a factor in affecting the sentiments in a review.

# Chapter-2
# Literature Review

This chapter highlights some of the previous studies in the field, citing existing gaps in knowledge on the how business benefit from the findings of sentiment analysis on customers.

## 2.1 Theoretical background of Online reviews

According to Sharma, Chakraborti, and Jha (2019), online shopping has gained global popularity over the past decade. This dramatic trend's primary reasons include the ease of internet access, availability of smartphones, increased awareness of e-commerce, and increased access to online shopping applications. Online shopping platforms such as Amazon enable customers to shop with convenience, save time, and get their products delivered at home within the shortest time possible. Competitive businesses prefer e-commerce over physical stores for the potential to reach more customers around the world (Sharma, Chakraborti, and Jha, 2019). E-commerce platforms also enable the company to save on enormous costs by setting up online stores. However, the customer demands on platforms such as Amazon on price and quality are drastically changing. The majority of customers are focusing more attention on high-quality and affordable products and services. In that regard, businesses must analyze the sentiments of customers to address these demands effectively.

Meire et al. (2019) argued that social media is a trending platform for markets to drive customer engagement today. Customer engagement initiatives enable businesses to boost their emotional bonds with customers located in different parts of the world. Product reviews also shape customer engagement (Nandal, Tanwar, and Pruthi, 2020). An analysis of the product reviews enables the business to gain insights into how customers feel about their products. According to Schoenmueller, Netzer, and Stahl (2020), online consumer reviews play a critical role in shaping customers' purchasing decisions online. A business must concentrate on analyzing and understanding these reviews to succeed in today's business environment.

Presently, businesses conduct sentiment analysis to enhance their competitiveness in the market. Karamitsos et al. (2019) argued that sentiment analysis enables companies to understand their products and services' views and experiences. As a result, companies can effectively design their marketing campaigns (Dong et al., 2017). Sentiment analysis also allows modern businesses to maximize the word-of-mouth marketing strategy. Competitive companies have gone the extra mile to utilize text mining methods to understand customer experiences better. Text mining enables the business to extract useful information from social media platforms, articles, and other sources. Zhao (2013) explained how to extract text from tweets using the code userTimeline (). Thus, a combination of sentiment analysis, text mining, and more methods plays a vital role in ensuring that business understand and capitalize on customer reviews online.

According to Jain, Kumar, and Mahanti (2018), sentiment extraction was an effective method of understanding customer suppositions online. The information gathered from online platforms and product review sites enable a business to enhance their marketing strategies. The product reviews also inform and shape customer purchasing decisions (Jain, Kumar, and Mahanti, 2018). Competitive companies such as Amazon capitalize on the information in decision-making. Lim et al. (2019) asserted that US top retailer's bank on online product reviews to enhance their marketing campaigns and enhanced business processes (Jagdale, Shirsat, and Deshmukh, 2019). For instance, if a specific product receives many negative reviews, the company investigates the issue to address it immediately. If the negative reviews are linked to pricing or quality, the company ensures that the issue is solved immediately.

Schoenmueller, Netzer, and Stahl (2020) collected an extensive data set of more than 280 million reviews to study their distribution. The data was created by more than twenty-four million reviewers from twenty-five sites, including Amazon and Yelp. The reviews covered different products and services. The study found that most product reviews online are less polar and positively imbalanced (Schoenmueller, Netzer, and Stahl, 2020). The study also found that the distribution of product reviews for similar products varies from one platform to another (Vyas and Uma, 2019). The reviews' variation is linked determined by various factors, including the rating scale, the online platform's business model, and the frequency of reviews (Schoenmueller, Netzer, and Stahl, 2020). Hence, to succeed in capitalizing on online product reviews, companies such as Amazon must take advantage of these factors.

Moreover, it is profound to note that businesses maximize sentiment analysis to enhance business processes and improve customer retention. Govindaraj and Gopalakrishnan (2016) asserted that an analysis of product reviews enables a business to understand customer experiences. A customer can post a review to show whether they are satisfied or unsatisfied with a specific product or service. However, most of the product reviews fail to indicate the extent of customer satisfaction. As a result, Govindaraj and Gopalakrishnan (2016) conducted a study to categorize the extent of customer satisfaction based on online reviews. They created a method of categorizing customer satisfaction based on acoustic and linguistic features. They proposed a model of categorizing customer reviews as highly positive, positive, neutral, and highly negative (Govindaraj and Gopalakrishnan, 2016). This study's results are consistent with previous research conducted by Ghasemaghaei et al. (2018) on the impact of the length of reviews and online sentiments. Customers tend to concentrate on extensive and detailed product reviews before making the final purchase (Singla, Randhawa, and Jain, 2017). Therefore, for an accurate decision based on online product reviews, companies need to investigate the actual extent of customer satisfaction with their products and services.

Sharma, Chakraborti, and Jha (2019) conducted a study to investigate how online reviews drive book sales at Amazon. According to the study, customers consider online reviews to be a dependable source of information. Customers find reviews to be more accessible and detailed. The study found that online reviews significantly shape user experiences and product prices. The findings are consistent with a previous investigation conducted by Chong et al. (2016) on online reviews and sentiments. Another issue that the study explored is online review valance. Sharma, Chakraborti, and Jha (2019) note that online reviews' impact on sales is contradictory. Some studies found that online review valance significantly impacts sales, while others found minimal impact. The impact is also dependent on factors such as product categories and qualitative text features.

Du et al. (2019) conducted a study focused on 142.8 million customer reviews from Amazon. The study focused on identifying each review's helpfulness and unhelpfulness by analyzing the summary headline, comment on the product, and helpfulness information. To enhance the accuracy of the findings, the researchers filtered all blank and non-English product reviews. Only those with the highest votes were selected (Du et al., 2019). The study found that an analysis of online product reviews on Amazon plays a significant role in today's e-commerce. Helpful reviews provide detailed information regarding the specific product or service based on customer experience (Meenakshi, Intwala, and Sawant, 2020). The reviews with the highest votes revealed that customers depend on the information to make accurate purchasing decisions. The findings are consistent with Anh, Nagai, and Nguyen's (2019) investigation on how customer reviews influence online shopping. Positive product reviews encourage customers to gain more credibility for the products they plan to purchase online.

# Chapter-3

## 3.1 Data Understanding

This is a list of over 34,000 consumer reviews for Amazon products like the Kindle, Fire TV Stick, and more provided by Datafiniti's Product Database. The dataset includes 21 attributes like basic product information, rating, review text, and more for each product.

| | id | name | asins | brand | categories | keys | manufacturer | reviews.date |
|---|---|---|---|---|---|---|---|---|
| 0 | AVqkIhwDv8e3D1O-lebb | All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi,... | B01AHB9CN2 | Amazon | Electronics,iPad & Tablets,All Tablets,Fire Ta... | 841667104676,amazon/53004484,amazon/b01ahb9cn2... | Amazon | 2017-01-13T00:00:00.000Z |
| 1 | AVqkIhwDv8e3D1O-lebb | All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi,... | B01AHB9CN2 | Amazon | Electronics,iPad & Tablets,All Tablets,Fire Ta... | 841667104676,amazon/53004484,amazon/b01ahb9cn2... | Amazon | 2017-01-13T00:00:00.000Z |
| 2 | AVqkIhwDv8e3D1O-lebb | All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi,... | B01AHB9CN2 | Amazon | Electronics,iPad & Tablets,All Tablets,Fire Ta... | 841667104676,amazon/53004484,amazon/b01ahb9cn2... | Amazon | 2017-01-13T00:00:00.000Z |
| 3 | AVqkIhwDv8e3D1O-lebb | All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi,... | B01AHB9CN2 | Amazon | Electronics,iPad & Tablets,All Tablets,Fire Ta... | 841667104676,amazon/53004484,amazon/b01ahb9cn2... | Amazon | 2017-01-13T00:00:00.000Z |
| 4 | AVqkIhwDv8e3D1O-lebb | All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi,... | B01AHB9CN2 | Amazon | Electronics,iPad & Tablets,All Tablets,Fire Ta... | 841667104676,amazon/53004484,amazon/b01ahb9cn2... | Amazon | 2017-01-12T00:00:00.000Z |

| reviews.date | reviews.dateAdded | reviews.dateSeen | ... | reviews.doRecommend | reviews.id | reviews.numHelpful | reviews.rating |
|---|---|---|---|---|---|---|---|
| 2017-01-13T00:00:00.000Z | 2017-07-03T23:33:15Z | 2017-06-07T09:04:00.000Z,2017-04-30T00:45:00.000Z | ... | True | NaN | 0.0 | 5.0 |
| 2017-01-13T00:00:00.000Z | 2017-07-03T23:33:15Z | 2017-06-07T09:04:00.000Z,2017-04-30T00:45:00.000Z | ... | True | NaN | 0.0 | 5.0 |
| 2017-01-13T00:00:00.000Z | 2017-07-03T23:33:15Z | 2017-06-07T09:04:00.000Z,2017-04-30T00:45:00.000Z | ... | True | NaN | 0.0 | 5.0 |
| 2017-01-13T00:00:00.000Z | 2017-07-03T23:33:15Z | 2017-06-07T09:04:00.000Z,2017-04-30T00:45:00.000Z | ... | True | NaN | 0.0 | 4.0 |
| 2017-01-12T00:00:00.000Z | 2017-07-03T23:33:15Z | 2017-06-07T09:04:00.000Z,2017-04-30T00:45:00.000Z | ... | True | NaN | 0.0 | 5.0 |

| | reviews.sourceURLs | reviews.text | reviews.title | reviews.userCity | reviews.userProvince | reviews.username |
|---|---|---|---|---|---|---|
| | http://reviews.bestbuy.com/3545/5620406/review... | This product so far has not disappointed. My c... | Kindle | NaN | NaN | Adapter |
| | http://reviews.bestbuy.com/3545/5620406/review... | great for beginner or experienced person. Boug... | very fast | NaN | NaN | truman |
| | http://reviews.bestbuy.com/3545/5620406/review... | Inexpensive tablet for him to use and learn on... | Beginner tablet for our 9 year old son. | NaN | NaN | DaveZ |
| | http://reviews.bestbuy.com/3545/5620406/review... | I've had my Fire HD 8 two weeks now and I love... | Good!!! | NaN | NaN | Shacks |
| | http://reviews.bestbuy.com/3545/5620406/review... | I bought this for my grand daughter when she c... | Fantastic Tablet for kids | NaN | NaN | explore42 |

Before diving into data analysis, it is necessary to take a look at the dataset and clean data if necessary. For sentimental analysis, it needs more steps to prepare the text for later on analysis. Here is the workflow of data analysis by Python in this project.

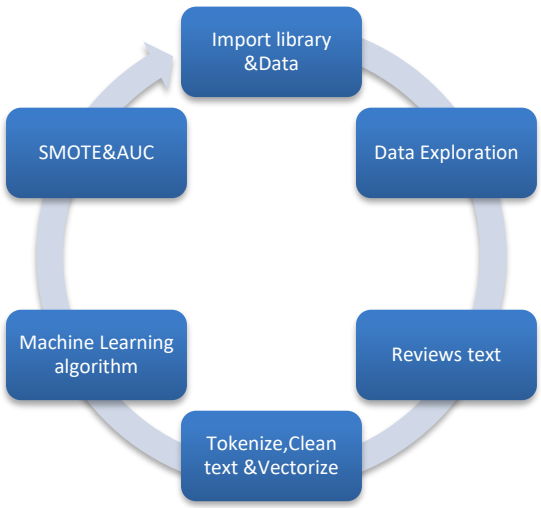



Figure: The workflow of the project

**Data exploration**

After importing the library and data, we can do some data exploration to generally understanding the dataset.

Based on the descriptive statistics below, we see the following:

- Average review score of 4.58, with low standard deviation Most review are positive from 2nd quartile onwards
- The average for number of reviews helpful (reviews.numHelpful) is 0.6 but high standard deviation The data are pretty spread out around the mean, and since can't have negative people finding something helpful, then this is only on the right tail side The range of most reviews will be between 0-13 people finding helpful (reviews.numHelpful)
- The most helpful review was helpful to 814 people This could be a detailed, rich review that will be worth looking at

```
data = df.copy()
data.describe()
```

| | reviews.id | reviews.numHelpful | reviews.rating | reviews.userCity | reviews.userProvince |
|---|---|---|---|---|---|
| count | 1.0 | 34131.000000 | 34627.000000 | 0.0 | 0.0 |
| mean | 111372787.0 | 0.630248 | 4.584573 | NaN | NaN |
| std | NaN | 13.215775 | 0.735653 | NaN | NaN |
| min | 111372787.0 | 0.000000 | 1.000000 | NaN | NaN |
| 25% | 111372787.0 | 0.000000 | 4.000000 | NaN | NaN |
| 50% | 111372787.0 | 0.000000 | 5.000000 | NaN | NaN |
| 75% | 111372787.0 | 0.000000 | 5.000000 | NaN | NaN |
| max | 111372787.0 | 814.000000 | 5.000000 | NaN | NaN |

Based on the below information :

- Drop reviews.userCity, reviews.userProvince, reviews.id, and reviews.didPurchase since these values are floats (for exploratory analysis only)
- Not every category have maximum number of values in comparison to total number of values
- reviews.text category has minimum missing data (34659/34660) -> Good news!
- We need to clean up the name column by referencing asins (unique products) since we have 7000 missing values

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34660 entries, 0 to 34659
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   id                    34660 non-null  object 
 1   name                  27900 non-null  object 
 2   asins                 34658 non-null  object 
 3   brand                 34660 non-null  object 
 4   categories            34660 non-null  object 
 5   keys                  34660 non-null  object 
 6   manufacturer          34660 non-null  object 
 7   reviews.date          34621 non-null  object 
 8   reviews.dateAdded     24039 non-null  object 
 9   reviews.dateSeen      34660 non-null  object 
 10  reviews.didPurchase   1 non-null      object 
 11  reviews.doRecommend   34066 non-null  object 
 12  reviews.id            1 non-null      float64
 13  reviews.numHelpful    34131 non-null  float64
 14  reviews.rating        34627 non-null  float64
 15  reviews.sourceURLs    34660 non-null  object 
 16  reviews.text          34659 non-null  object 
 17  reviews.title         34655 non-null  object 
 18  reviews.userCity      0 non-null      float64
 19  reviews.userProvince  0 non-null      float64
 20  reviews.username      34658 non-null  object 
dtypes: float64(5), object(16)
memory usage: 5.6+ MB
```

**Number of Unique Product** –

```
data["asins"].unique()
```

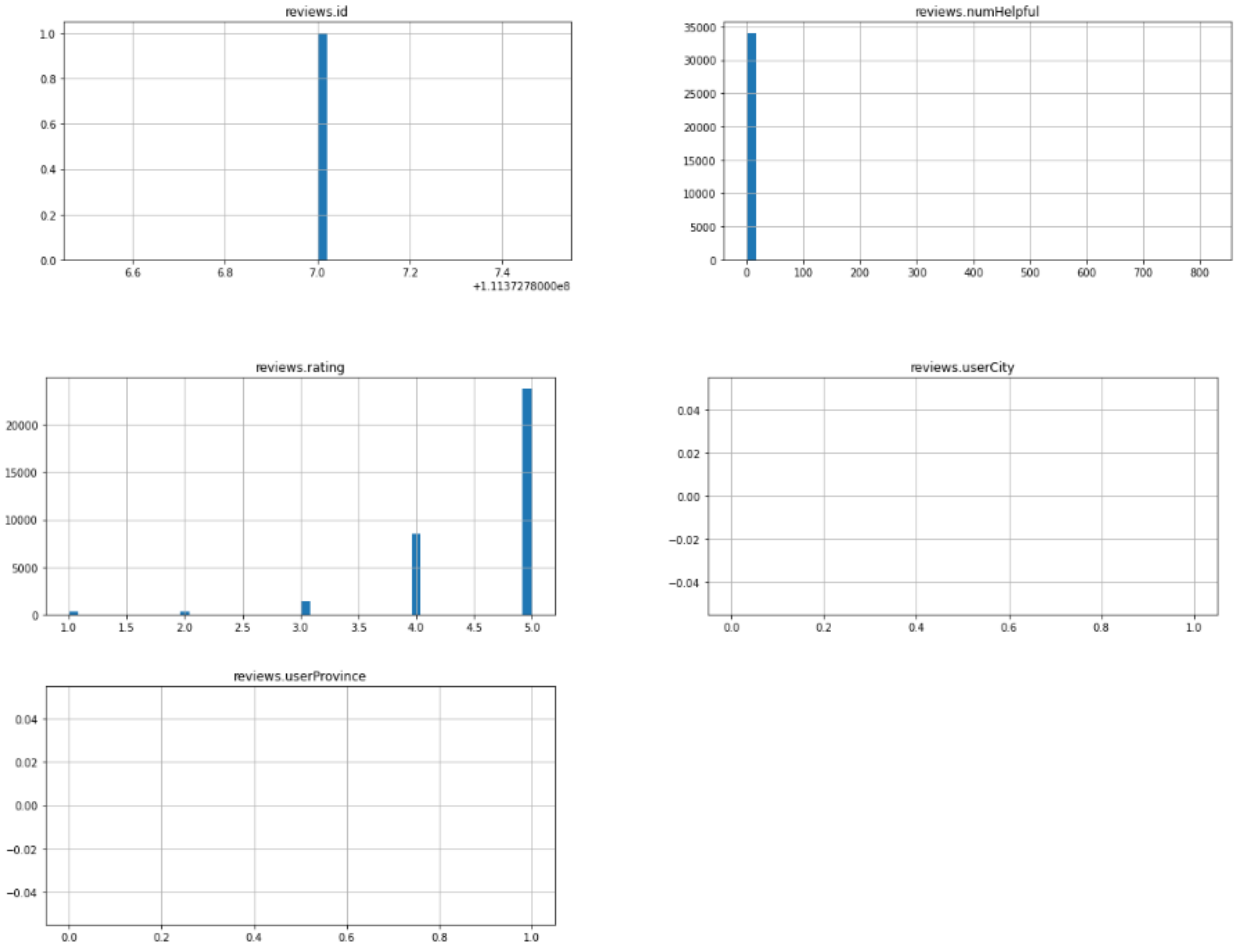
```
array(['B01AHB9CN2', 'B00VINDBJK', 'B005PB2T0S', 'B002Y27P3M',
       'B01AHB9CYG', 'B01AHB9C1E', 'B01J2G4VBG', 'B00ZV9PXP2',
       'B0083Q04TA', 'B018Y229OU', 'B00REQKWGA', 'B00IOYAM4I',
       'B018T075DC', nan, 'B00DU15MU4', 'B018Y225IA', 'B005PB2T2Q',
       'B018Y23MNM', 'B00OQVZDJM', 'B00IOY8XWQ', 'B00LO29KXQ',
       'B00QJDU3KY', 'B018Y22C2Y', 'B01BFIBRIE', 'B01J4ORNHU',
       'B018SZT3BK', 'B00UH4D8G2', 'B018Y22BI4', 'B00TSUGXKE',
       'B00L9EPT8O,B01E6AO69U', 'B018Y23P7K', 'B00X4WHP5E', 'B00QFQRELG',
       'B00LW9XOJM', 'B00QL1ZN3G', 'B0189XYY0Q', 'B01BH83OOM',
       'B00BFJAHF8', 'B00U3FPN4U', 'B002Y27P6Y', 'B006GWO5NE',
       'B006GWO5WK'], dtype=object)
```

```
asins_unique = len(data["asins"].unique())
print("Number of Unique ASINs: " + str(asins_unique))
```

```
Number of Unique ASINs: 42
```

**Visualizing the distributions of numerical variables:**

```
: data.hist(bins=50, figsize=(20,15)) # builds histogram and set the number of bins and fig size (width, height)
  plt.show()
```




Based on the distributions above:

- reviews.numHelpful: Outliers in this case are valuable, so we may want to weight reviews that had more than 50+ people who find them helpful
- reviews.rating: Majority of examples were rated highly (looking at rating distribution). There is twice amount of 5 star ratings than the others ratings combined

## 3.2 Split into Training Dataset

- Before we explore the dataset we're going to split it into training set and test sets
- Our goal is to eventually train a sentiment analysis classifier
- Since the majority of reviews are positive (5 stars), we will need to do a stratified split on the reviews score to ensure that -we don't train the classifier on imbalanced data
- To use sklearn's Stratified Shuffle Split class, we're going to remove all samples that have NAN in review score, then covert all review scores to integer data type

```
reviews = strat_train.copy()
reviews.head(2)
```

| | id | name | asins | brand | categories | keys | manufactur |
|---|---|---|---|---|---|---|---|
| 26302 | AVpfl8cLLJeJML43AE3S | Amazon Fire Tv,,,\nAmazon Fire Tv,,, | B00L9EPT8O,B01E6AO69U | Amazon | Stereos,Remote Controls,Amazon Echo,Audio Dock... | echowhite/263039693056,echowhite/152558276095,... | Amazc |
| 24615 | AVpfl8cLLJeJML43AE3S | Echo (White),,,\nEcho (White),,, | B00L9EPT8O,B01E6AO69U | Amazon | Stereos,Remote Controls,Amazon Echo,Audio Dock... | echowhite/263039693056,echowhite/152558276095,... | Amazc |

2 rows × 21 columns

**Exploring the names / ASINs column**

```
len(reviews["name"].unique()), len(reviews["asins"].unique())
```

```
(47, 36)
```

```
reviews.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 27701 entries, 26302 to 9892
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   id                    27671 non-null  object
 1   name                  22285 non-null  object
 2   asins                 27669 non-null  object
 3   brand                 27671 non-null  object
 4   categories            27671 non-null  object
 5   keys                  27671 non-null  object
 6   manufacturer          27671 non-null  object
 7   reviews.date          27654 non-null  object
 8   reviews.dateAdded     19189 non-null  object
 9   reviews.dateSeen      27671 non-null  object
 10  reviews.didPurchase   1 non-null      object
 11  reviews.doRecommend   27257 non-null  object
 12  reviews.id            1 non-null      float64
 13  reviews.numHelpful    27306 non-null  float64
 14  reviews.rating        27671 non-null  float64
 15  reviews.sourceURLs    27671 non-null  object
 16  reviews.text          27670 non-null  object
 17  reviews.title         27666 non-null  object
 18  reviews.userCity      0 non-null      float64
 19  reviews.userProvince  0 non-null      float64
 20  reviews.username      27669 non-null  object
dtypes: float64(5), object(16)
memory usage: 4.6+ MB
```

Working hypothesis: there are only 35 products based on the training data ASINs

- One for each ASIN, but more product names (47)
- ASINs are what's important here since we're concerned with products. There's a one to many relationship between ASINs and names
- A single ASIN can have many names due to different vendor listings
- There could also a lot of missing names/more unique names with slight variations in title (i.e. 8gb vs 8 gb, NAN for product names)

```
reviews.groupby("asins")["name"].unique()
```

```
asins
B005PB2T0S              [Kindle Keyboard,,,\nKindle Keyboard,,,, Amazo...
B005PB2T2Q              [Fire Kids Edition Tablet, 7 Display, Wi-Fi, 1...
B00DU15MJ4              [Brand New Amazon Kindle Fire 16gb 7 Ips Displ...
B00IOY8XWQ              [Kindle Voyage E-reader, 6 High-Resolution Dis...
B00IOYAM4I              [Brand New Amazon Kindle Fire 16gb 7 Ips Displ...
B00L9EPT8O,B01E6AO69U   [Amazon Fire Tv,,,\nAmazon Fire Tv,,,, Echo (W...
B00LO29KXQ              [Fire Tablet, 7 Display, Wi-Fi, 8 GB - Include...
B00LW9XOJM                                                      [nan]
B00OQVZDJM              [Amazon Kindle Paperwhite - eBook reader - 4 G...
B00QFQRELG                                                      [nan]
B00QJDU3KY              [Fire Tablet, 7 Display, Wi-Fi, 8 GB - Include...
B00QL1ZN3G                                                      [nan]
B00REQKWGA              [Brand New Amazon Kindle Fire 16gb 7 Ips Displ...
B00TSUGXKE                        [Echo (White),,,\nEcho (White),,,]
B00U3FPN4U                                                      [nan]
B00UH4D8G2                        [Echo (White),,,\nEcho (White),,,]
B00VINDBJK              [Kindle Oasis E-reader with Leather Charging C...
B00X4WHP5E                                                      [nan]
B00ZV9PXP2              [Fire Tablet, 7 Display, Wi-Fi, 8 GB - Include...
B0189XYY0Q                                                      [nan]
B018SZT3BK              [Fire Tablet, 7 Display, Wi-Fi, 8 GB - Include...
B018T075DC              [Brand New Amazon Kindle Fire 16gb 7 Ips Displ...
B018Y225IA              [Brand New Amazon Kindle Fire 16gb 7 Ips Displ...
B018Y229OU              [Fire Tablet, 7 Display, Wi-Fi, 8 GB - Include...
B018Y22BI4                        [Echo (White),,,\nEcho (White),,,]
B018Y22C2Y              [Fire Tablet, 7 Display, Wi-Fi, 8 GB - Include...
B018Y23MNM              [Fire Kids Edition Tablet, 7 Display, Wi-Fi, 1...
B018Y23P7K                                                      [nan]
B01AHB9C1E              [Fire HD 8 Tablet with Alexa, 8 HD Display, 32...
B01AHB9CN2              [All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi...
B01AHB9CYG              [All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi...
B01BFIBRIE              [Fire Tablet, 7 Display, Wi-Fi, 8 GB - Include...
B01BH83OOM                                                      [nan]
B01J2G4VBG              [Amazon 5W USB Official OEM Charger and Power ...
B01J4ORNHU              [Fire Tablet, 7 Display, Wi-Fi, 8 GB - Include...
Name: name, dtype: object
```

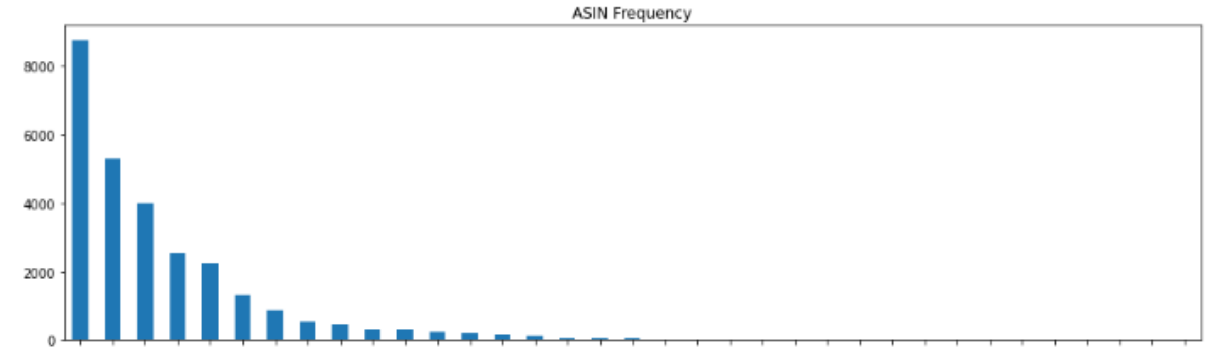
```
reviews[reviews["asins"] == "B00L9EPT8O,B01E6AO69U"]["name"].value_counts()
```

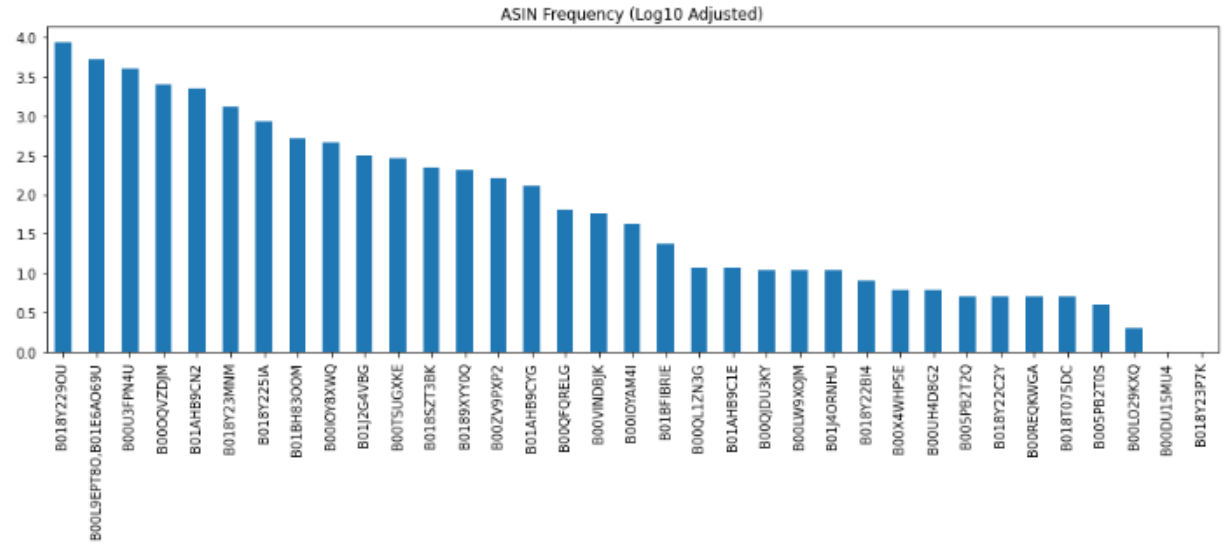
```
Echo (White),,,\nEcho (White),,,
2309
Amazon Fire Tv,,,\nAmazon Fire Tv,,,
2027
Amazon - Amazon Tap Portable Bluetooth and Wi-Fi Speaker - Black,,,\nAmazon - Amazon Tap Portable Bluetooth and Wi-Fi Speaker -
Black,,,                                                                                                                  245
Amazon Fire Hd 10 Tablet, Wi-Fi, 16 Gb, Special Offers - Silver Aluminum,,,\nAmazon Fire Hd 10 Tablet, Wi-Fi, 16 Gb, Special Of
fers - Silver Aluminum,,,                                                                                                  92
Amazon 9W PowerFast Official OEM USB Charger and Power Adapter for Fire Tablets and Kindle eReaders,,,\nAmazon 9W PowerFast Off
icial OEM USB Charger and Power Adapter for Fire Tablets and Kindle eReaders,,,      31
Kindle Dx Leather Cover, Black (fits 9.7 Display, Latest and 2nd Generation Kindle Dxs),,
7
Amazon 5W USB Official OEM Charger and Power Adapter for Fire Tablets and Kindle eReaders,,,\nAmazon 5W USB Official OEM Charge
r and Power Adapter for Fire Tablets and Kindle eReaders,,,                                                                6
New Amazon Kindle Fire Hd 9w Powerfast Adapter Charger + Micro Usb Angle Cable,,,\nNew Amazon Kindle Fire Hd 9w Powerfast Adapt
er Charger + Micro Usb Angle Cable,,,                                                                                      6
Amazon Fire Hd 6 Standing Protective Case(4th Generation - 2014 Release), Cayenne Red,,,\nAmazon Fire Hd 6 Standing Protective
Case(4th Generation - 2014 Release), Cayenne Red,,,                                                                        4
Amazon Kindle Fire 5ft USB to Micro-USB Cable (works with most Micro-USB Tablets),,,\nAmazon Kindle Fire 5ft USB to Micro-USB C
able (works with most Micro-USB Tablets),,,                                                                               3
Echo (Black),,,\nEcho (Black),,,
3
Amazon Fire Hd 6 Standing Protective Case(4th Generation - 2014 Release), Cayenne Red,,,\nAmazon 5W USB Official OEM Charger an
d Power Adapter for Fire Tablets and Kindle eReaders,,,                                                                    1
New Amazon Kindle Fire Hd 9w Powerfast Adapter Charger + Micro Usb Angle Cable,,,\n
1
Amazon Fire Tv,,,\nKindle Dx Leather Cover, Black (fits 9.7 Display, Latest and 2nd Generation Kindle Dxs)",,
1
Echo (Black),,,\nAmazon 9W PowerFast Official OEM USB Charger and Power Adapter for Fire Tablets and Kindle eReaders,,,
1
Coconut Water Red Tea 16.5 Oz (pack of 12),,,\nAmazon Fire Tv,,,
1
Name: name, dtype: int64
```

Confirmed our hypothesis that each ASIN can have multiple names.

Therefore we should only really concern ourselves with which ASINs do well, not the product names.

```
fig = plt.figure(figsize=(16,10))
ax1 = plt.subplot(211)
ax2 = plt.subplot(212, sharex = ax1)
reviews["asins"].value_counts().plot(kind="bar", ax=ax1, title="ASIN Frequency")
np.log10(reviews["asins"].value_counts()).plot(kind="bar", ax=ax2, title="ASIN Frequency (Log10 Adjusted)")
plt.show()
```





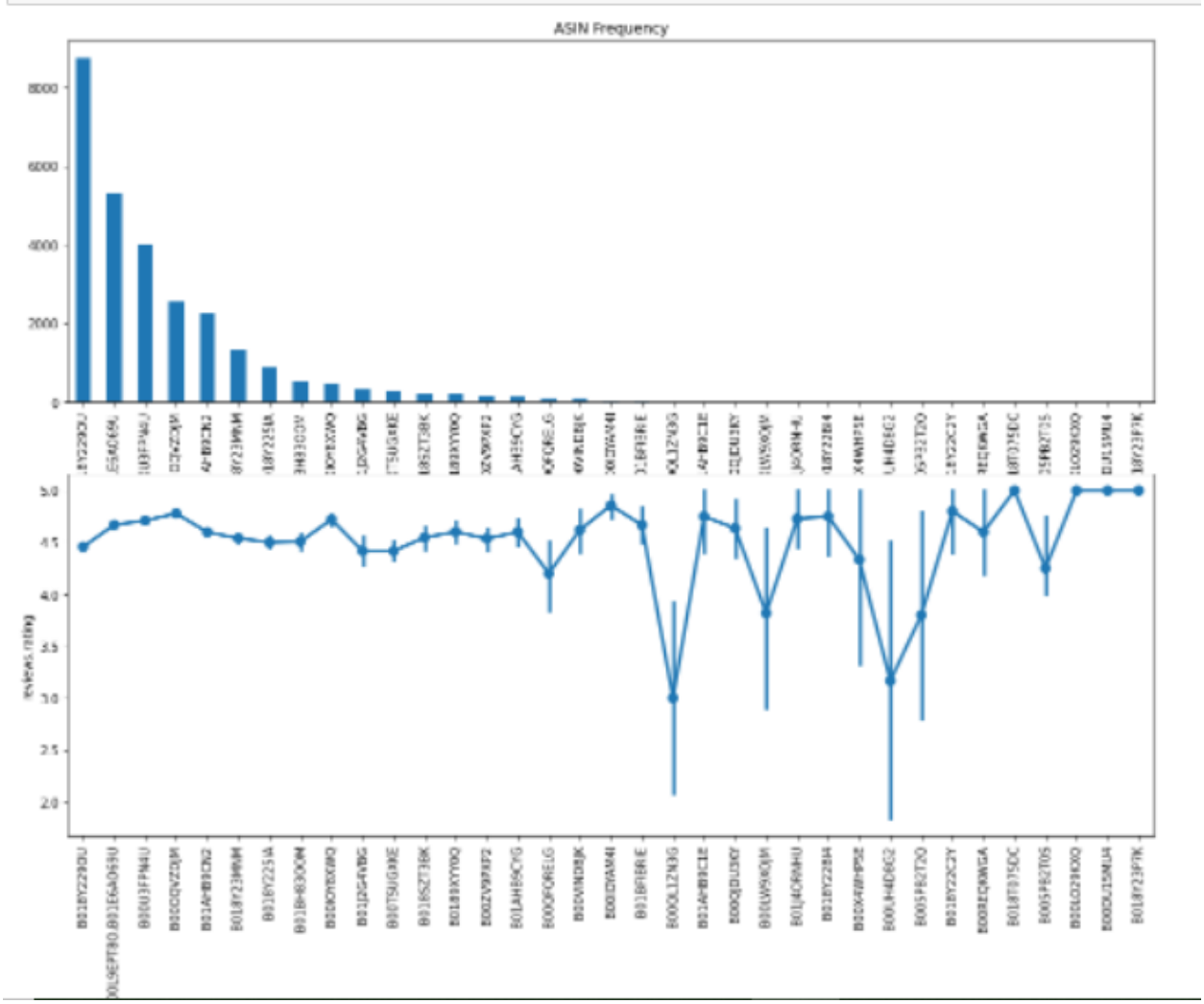

- Based on the bar graph for ASINs, we see that certain products have significantly more reviews than other products, which may indicate a higher sale in those specific products
- We also see that the ASINs have a "right tailed" distribution which can also suggest that certain products have higher sales which can correlate to the higher ASINs frequencies in the reviews
- We also took the log of the ASINs to normalize the data, in order display an in-depth picture of each ASINs, and we see that the distribution still follows a "right tailed" distribution

```
# Entire training dataset average rating
reviews["reviews.rating"].mean()
```

```
4.58819702938094
```

**Exploring the reviews.rating / ASINs column**

```
asins_count_ix = reviews["asins"].value_counts().index
plt.subplots(2,1,figsize=(16,12))
plt.subplot(2,1,1)
reviews["asins"].value_counts().plot(kind="bar", title="ASIN Frequency")
plt.subplot(2,1,2)
sns.pointplot(x="asins", y="reviews.rating", order=asins_count_ix, data=reviews)
plt.xticks(rotation=90)
plt.show()
```

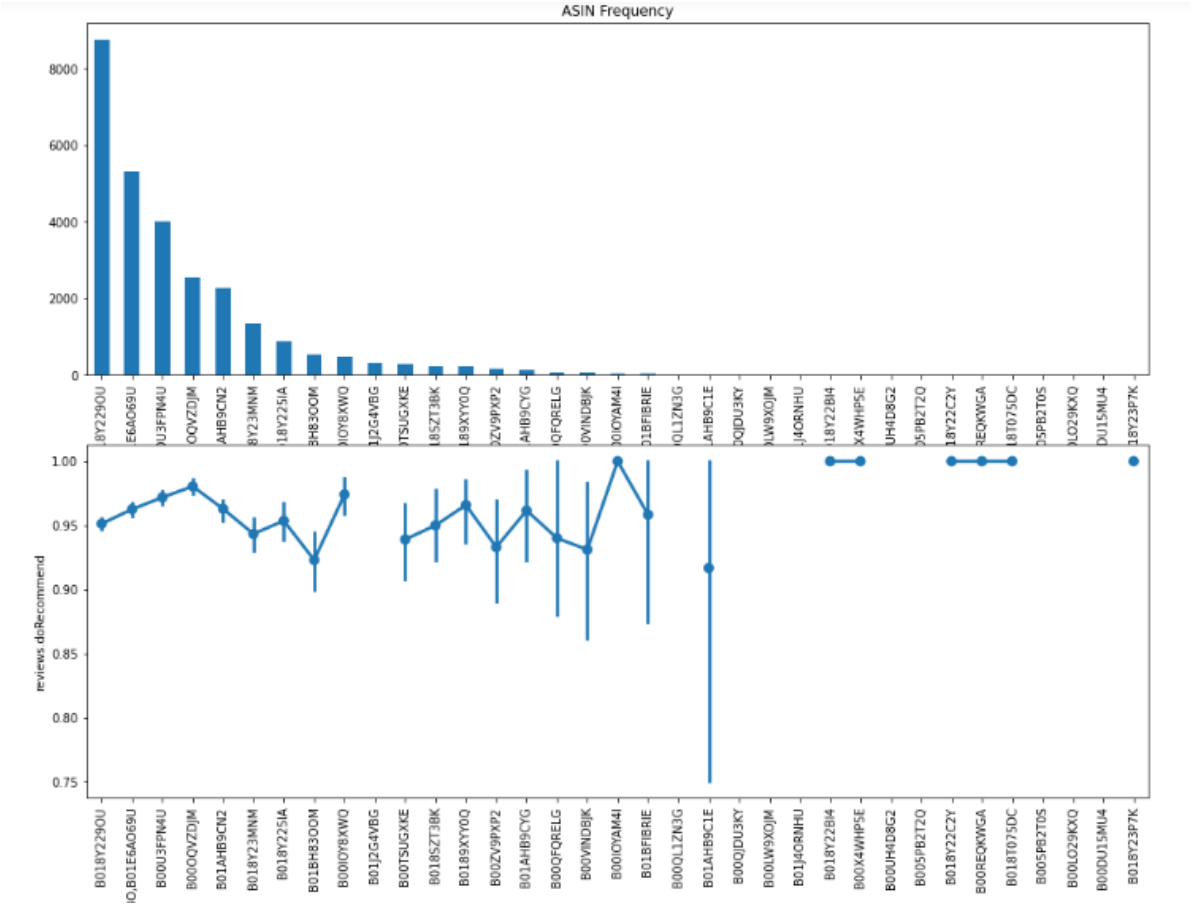



- 1a) The most frequently reviewed products have their average review ratings in the 4.5 - 4.8 range, with little variance
- 1b) Although there is a slight inverse relationship between the ASINs frequency level and average review ratings for the first 4 ASINs, this relationship is not significant since the average review for the first 4 ASINs are rated between 4.5 - 4.8, which is considered good overall reviews
- 2a) For ASINs with lower frequencies as shown on the bar graph (top), we see that their corresponding average review ratings on the point-plot graph (bottom) has significantly higher variance as shown by the length of the vertical lines. As a result, we suggest that, the average review ratings for ASINs with lower frequencies are not significant for our analysis due to high variance
- 2b) On the other hand, due to their lower frequencies for ASINs with lower frequencies, we suggest that this is a result of lower quality products

- 2c) Furthermore, the last 4 ASINs have no variance due to their significantly lower frequencies, and although the review ratings are a perfect 5.0, but we should not consider the significance of these review ratings due to lower frequency as explained in 2(a).

**Explorations reviews.doRecommend / ASINs column**

```
plt.subplots (2,1,figsize=(16,12))
plt.subplot(2,1,1)
reviews["asins"].value_counts().plot(kind="bar", title="ASIN Frequency")
plt.subplot(2,1,2)
sns.pointplot(x="asins", y="reviews.doRecommend", order=asins_count_ix, data=reviews)
plt.xticks(rotation=90)
plt.show()
```




- From this analysis, we can see that the first 19 ASINs show that consumers recommend the product, which is consistent with the "reviews.rating / ASINs" analysis above, where the first 19 ASINs have good ratings between 4.0 to 5.0
- The remaining ASINs have fluctuating results due to lower sample size, which should not be considered

Note: reviews.text will be analyzed in Sentiment Analysis.

## 3.3 Correlations

Correlation analysis is a statistical method used to measure the strength of the linear relationship between two variables and compute their association. Correlation analysis calculates the level of change in one variable due to the change in the other. A high correlation points to a strong relationship between the two variables, while a low correlation means that the variables are weakly related. Correlation is a vicariate analysis that measures the strength of association between two variables and the direction of the relationship. In terms of the strength of the relationship, the correlation coefficient's value varies between +1 and -1. A value of ± 1 indicates a perfect degree of association between the two variables.

In [28]:
```
corr_matrix = reviews.corr()
corr_matrix
# Here we can analyze reviews.ratings with asins
```

Out[28]:

| | reviews.id | reviews.numHelpful | reviews.rating | reviews.userCity | reviews.userProvince |
|---|---|---|---|---|---|
| reviews.id | NaN | NaN | NaN | NaN | NaN |
| reviews.numHelpful | NaN | 1.000000 | -0.045019 | NaN | NaN |
| reviews.rating | NaN | -0.045019 | 1.000000 | NaN | NaN |
| reviews.userCity | NaN | NaN | NaN | NaN | NaN |
| reviews.userProvince | NaN | NaN | NaN | NaN | NaN |

In [30]:
```
counts = reviews["asins"].value_counts().to_frame()
counts.head()
```

Out[30]:

| | asins |
|---|---|
| B018Y229OU | 8759 |
| B00L9EPT8O,B01E6AO69U | 5293 |
| B00U3FPN4U | 4009 |
| B00OQVZDJM | 2554 |
| B01AHB9CN2 | 2260 |

In [31]:
```
avg_rating = reviews.groupby("asins")["reviews.rating"].mean().to_frame()
avg_rating.head()
```

Out[31]:

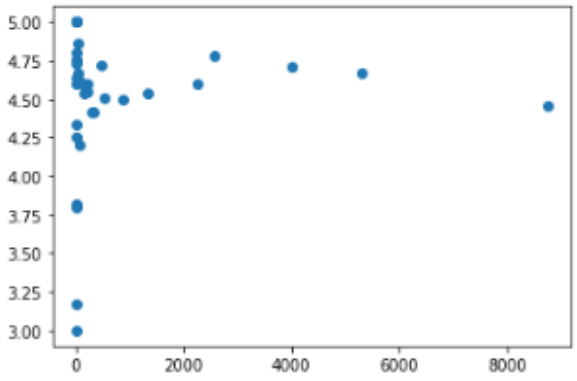
| | reviews.rating |
|---|---|
| asins | |
| B005PB2T0S | 4.250000 |
| B005PB2T2Q | 3.800000 |
| B00DU15MU4 | 5.000000 |
| B00IOY8XWQ | 4.721382 |
| B00IOYAM4I | 4.857143 |

```
table = counts.join(avg_rating)
table.head(30)
```

| | asins | reviews.rating |
|---|---|---|
| B018Y225OU | 8759 | 4.457358 |
| B00L9EPT8O,B01E6AO69U | 5293 | 4.668997 |
| B00U3FPN4U | 4009 | 4.710152 |
| B00OQVZDJM | 2554 | 4.778387 |
| B01AHB9CN2 | 2260 | 4.598673 |
| B018Y23MNM | 1323 | 4.539683 |
| B018Y225IA | 861 | 4.500581 |
| B01BH83OOM | 521 | 4.508637 |
| B00IOY8XWQ | 463 | 4.721382 |
| B01J2G4VBG | 316 | 4.417722 |
| B00TSUGXKE | 295 | 4.416949 |
| B018SZT3BK | 221 | 4.547511 |
| B018SXYY0Q | 206 | 4.601942 |
| B00ZV9PXP2 | 164 | 4.536585 |
| B01AHB9CYG | 130 | 4.600000 |
| B00QFQRELG | 65 | 4.200000 |
| B00VINDBJK | 58 | 4.620690 |
| B00IOYAM4I | 42 | 4.857143 |
| B01BFIBRIE | 24 | 4.666667 |
| B00QL1ZN3G | 12 | 3.000000 |
| B01AHB9C1E | 12 | 4.750000 |
| B00QJDU3KY | 11 | 4.636364 |
| B00LW9XOJM | 11 | 3.818182 |
| B01J4ORNHU | 11 | 4.727273 |
| B018Y22BI4 | 8 | 4.750000 |
| B00X4WHP5E | 6 | 4.333333 |
| B00UH4D8G2 | 6 | 3.166667 |
| B005PB2T2Q | 5 | 3.800000 |
| B018Y22C2Y | 5 | 4.800000 |
| B00REQKWGA | 5 | 4.600000 |

```
plt.scatter("asins", "reviews.rating", data=table)
table.corr()
```

| | asins | reviews.rating |
|---|---|---|
| asins | 1.000000 | 0.084919 |
| reviews.rating | 0.084919 | 1.000000 |




From our analysis in data exploration above between ASINs and reviews.rating, we discovered that there are many ASINs with low occurrence that have high variances, as a result we concluded that theses low occurrence ASINs are not significant in our analysis given the low sample size.

Similarly in our correlation analysis between ASINs and reviews.rating, we see that there is almost no correlation which is consistent with our findings.

## 3.4 Sentiment Analysis

Sentiment analysis (or opinion mining) is a natural language processing (NLP) technique used to determine whether data is positive, negative or neutral. Sentiment analysis is often performed on textual data to help businesses monitor brand and product sentiment in customer feedback, and understand customer needs.

**Importance**

Since humans express their thoughts and feelings more openly than ever before, sentiment analysis is fast becoming an essential tool to monitor and understand sentiment in all types of data.

Automatically analyzing customer feedback, such as opinions in survey responses and social media conversations, allows brands to learn what makes customers happy or frustrated, so that they can tailor products and services to meet their customers' needs.

For example, using sentiment analysis to automatically analyze 4,000+ open-ended responses in your customer satisfaction surveys could help you discover why customers are happy or unhappy at each stage of the customer journey.

Maybe you want to track brand sentiment so you can detect disgruntled customers immediately and respond as soon as possible. Maybe you want to compare sentiment from one quarter to the next to see if you need to take action. Then you could dig deeper into your qualitative data to see why sentiment is falling or rising.

### Set Target Variable (Sentiments)

```
# Segregate ratings from 1-5 into positive, neutral, and negative.
```

```
def sentiments(rating):
    if (rating == 5) or (rating == 4):
        return "Positive"
    elif rating == 3:
        return "Neutral"
    elif (rating == 2) or (rating == 1):
        return "Negative"
# Add sentiments to the data
strat_train["Sentiment"] = strat_train["reviews.rating"].apply(sentiments)
strat_test["Sentiment"] = strat_test["reviews.rating"].apply(sentiments)
strat_train["Sentiment"][:20]
```

```
Out[36]: 26302    Positive
         24615    Positive
         591      Positive
         8698     Positive
         30784    Positive
         7564     Positive
         14538    Positive
         5744     Positive
         19111    Positive
         27154    Positive
         31778    Positive
         27408     Neutral
         22776    Positive
         21619    Positive
         4274      Neutral
         18606    Positive
         4383     Positive
         30669    Positive
         28475    Positive
         19311    Positive
         Name: Sentiment, dtype: object
```

```
In [37]:
# Prepare data
X_train = strat_train["reviews.text"]
X_train_targetSentiment = strat_train["Sentiment"]
X_test = strat_test["reviews.text"]
X_test_targetSentiment = strat_test["Sentiment"]
print(len(X_train), len(X_test))
```

```
27701 6926
```

```
In [38]: # 27,701 training samples and 6926 testing samples.
```

## Extract Features

Here we will turn content into numerical feature vectors using the Bag of Words strategy:

- Assign fixed integer id to each word occurrence (integer indices to word occurrence dictionary) X[i,j] where i is the integer indices, j is the word occurrence, and X is an array of words (our training set)
- In order to implement the Bag of Words strategy, we will use SciKit-Learn's CountVectorizer to performs the following:
- Text preprocessing:
    - - Tokenization (breaking sentences into words)

      - Stopwords (filtering "the", "are", etc)
- Occurrence counting (builds a dictionary of features from integer indices with word occurrences)
- Feature Vector (converts the dictionary of text documents into a feature vector)

**Tokenization**

Tokenization is toseparate word-like units from text(Grefenstette and Tapanainen, 1994), which is also known as word segmentation(Palmer, n.d., p. 2).When doing text mining, the first preprocessing is word segmentation. English words are naturally separated by Spaces and can be easily separated by Spaces, but sometimes multiple words need to be treated as one word. For example, "New Jersey ", need to be looked as one word.In Chinese, participles are a special problem to be solved because there is no space. Whether in English or Chinese, the principle of word segmentation is similar.Modern participles are all based on statistical participles, and statistical sample content comes from some standard corpus(Liu, 2017a).

**Vectorization**

In the step of counting the word frequency, we will get the word frequency of all the words in the text. With the word frequency, we can use the word vector to represent the text(Liu, 2017b).The vectorization method is easy to use and straightforward, but it is difficult to use in some scenarios such as the vocabulary after word segmentation is very large. Hash Trick is a commonly used method to reduce the dimension of text features.

```
: # Replace "nan" with space
  X_train = X_train.fillna(' ')
  X_test = X_test.fillna(' ')
  X_train_targetSentiment = X_train_targetSentiment.fillna(' ')
  X_test_targetSentiment = X_test_targetSentiment.fillna(' ')

  # Text preprocessing and occurance counting
  from sklearn.feature_extraction.text import CountVectorizer
  count_vect = CountVectorizer()
  X_train_counts = count_vect.fit_transform(X_train)
  X_train_counts.shape
: (27701, 12497)
```

Here we have 27,701 training samples and 12,526 distinct words in our training sample.

Also, with longer documents, we typically see higher average count values on words that carry very little meaning, this will overshadow shorter documents that have lower average counts with same frequencies, as a result, we will use TfidfTransformer to reduce this redundancy:

- Term Frequencies (Tf) divides number of occurrences for each word by total number of words
- Term Frequencies times Inverse Document Frequency (Tfidf) downscales the weights of each word (assigns less value to unimportant stop words ie. "the", "are", etc)

```
from sklearn.feature_extraction.text import TfidfTransformer
tfidf_transformer = TfidfTransformer(use_idf=False)
X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
X_train_tfidf.shape
```

```
(27701, 12497)
```

## 3.5 Building a Pipeline from the Extracted Features

A **data pipeline** is a series of data processing steps. If the data is not currently loaded into the data platform, then it is ingested at the beginning of the pipeline. Then there are a series of steps in which each step delivers an output that is the input to the next step. This continues until the pipeline is complete. In some cases, independent steps may be run in parallel.

I am using Multinominal Naive Bayes as my Classifier

- Multinominal Niave Bayes is most suitable for word counts where data are typically represented as **word vector counts** (number of times outcome number X[i,j] is observed over the n trials), while also ignoring non-occurrences of a feature i
- Naive Bayes is a simplified version of Bayes Theorem, where all features are assumed conditioned independent to each other (the classifiers), P(x|y) where x is the feature and y is the classifier.

### 3.5.1. Multinomial Naive Bayes

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 greatest chance. The Naive Bayes classifier is made up of a number of algorithms that all have one thing in common: each feature being classed is unrelated to any other feature. A feature's existence or absence has no bearing on the inclusion or exclusion of another feature.

**Working Procedure –**

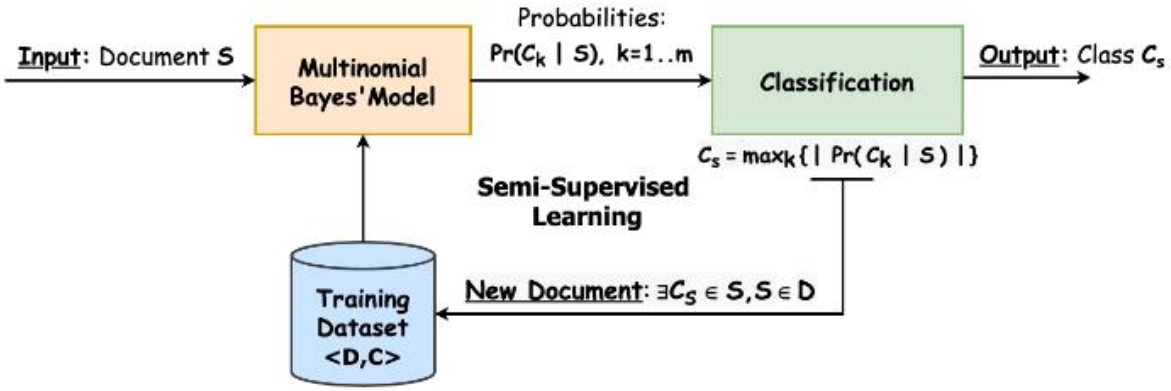
The Naive Bayes method is a strong tool for analyzing text input and solving problems with numerous classes. Because the Naive Bayes theorem is based on the Bayes theorem, it is necessary to first comprehend the Bayes theorem notion. The Bayes theorem, which was developed by Thomas Bayes, estimates the likelihood of occurrence based on prior knowledge of the event's conditions. When predictor B itself is available, we calculate the likelihood of class A. It's based on the formula below:

$$P(A|B) = P(A) * P(B|A)/P(B).$$

It is simple to implement because all you have to do is calculate probability. This approach works with both continuous and discrete data. It's straightforward and can be used to forecast real-time applications. It's very scalable and can handle enormous datasets with ease.

This algorithm's prediction accuracy is lower than that of other probability algorithms. It isn't appropriate for regression. The Naive Bayes technique can only be used to classify textual input and cannot be used to estimate numerical values.

### The semi-supervised learning process:




The process, illustrated above, provides an ability to perform classification, assigning samples to a finite set of classes, similar to using the expectation-maximization algorithm (EM).

The multinomial Bayesian model is an efficient alternative to the known K-means clustering and decision trees algorithms, capable of classifying various data that are normally not easy to be quantified. For example, it can be used as part of ANN-based text classification models that learn and predict texts to classes, based on the summary of their contents, inferred by the multinomial Bayesian classifier.

Python Code-

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline
clf_multiNB_pipe = Pipeline([("vect", CountVectorizer()), ("tfidf", TfidfTransformer()), ("clf_nominalNB", MultinomialNB())])
clf_multiNB_pipe.fit(X_train, X_train_targetSentiment)
```

```
Pipeline(steps=[('vect', CountVectorizer()), ('tfidf', TfidfTransformer()),
                ('clf_nominalNB', MultinomialNB())])
```

```
In [44]: import numpy as np
         predictedMultiNB = clf_multiNB_pipe.predict(X_test)
         np.mean(predictedMultiNB == X_test_targetSentiment)*100
```

```
Out[44]: 93.21397632110886
```

## 3.5.2. Logistic Regression

In statistics, the (binary) **logistic model** (or **logit model**) is a statistical model that models the probability of one event (out of two alternatives) taking place by having the log-odds (the logarithm of the odds) for the event be a linear combination of one or more independent variables ("predictors"). In regression analysis, **logistic regression**[1] (or **logit regression**) is estimating the parameters of a logistic model (the coefficients in the linear combination). Formally, in binary logistic regression there is a single binary dependent variable, coded by a indicator variable, where the two values are labeled "0" and "1", while the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The corresponding probability of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labeling; the function that converts log-odds to probability is the logistic function, hence the name. The unit of measurement for the log-odds scale is called a *logit*, from ***log****istic un****it***, hence the alternative names. Background and Definition for formal mathematics, and Example for a worked example.

Binary variables are widely used in statistics to model the probability of a certain class or event taking place, such as the probability of a team winning, of a patient being healthy, etc. (see Applications), and the logistic model has been the most commonly used model for binary regression since about 1970. Binary variables can be generalized to categorical variables when there are more than two possible values (e.g. whether an image is of a cat, dog, lion, etc.), and the binary logistic regression generalized to multinomial logistic regression. If the multiple categories are ordered, one can use the ordinal logistic regression (for example the proportional odds ordinal logistic model). See Extensions for further extensions. The logistic regression model itself simply models probability of output in terms of input and does not perform statistical classification (it is not a classifier), though it can be used to make a classifier, for instance by choosing a cutoff value and classifying inputs with probability greater than the cutoff as one class, below the cutoff as the other; this is a common way to make a binary classifier.

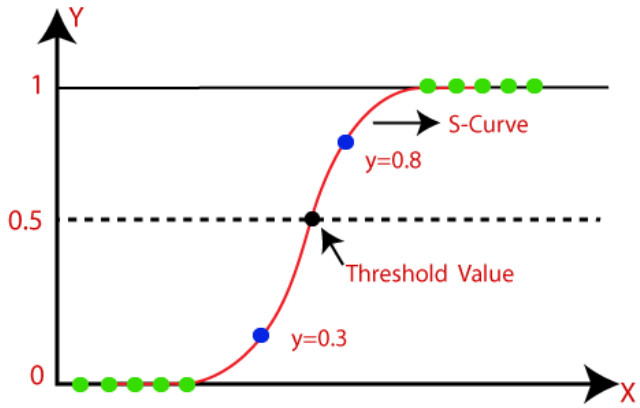
Analogous linear models for binary variables with a different sigmoid function instead of the logistic function (to convert the linear combination to a probability) can also be used, most notably the probit model; see Alternatives. The defining characteristic of the logistic model is that increasing one of the independent variables multiplicatively scales the odds of the given outcome at a *constant* rate, with each independent variable having its own parameter; for a binary dependent variable this generalizes the odds ratio. More abstractly, the logistic function is the natural parameter for the Bernoulli distribution, and in this sense is the "simplest" way to convert a real number to a probability. In particular, it maximizes entropy (minimizes added information), and in this sense makes the fewest assumptions of the data being modeled;

Logistic regression is used in various fields, including machine learning, most medical fields, and social sciences. For example, the Trauma and Injury Severity Score (TRISS), which is widely used to predict mortality in injured patients, was originally developed by Boyd *et al.* using logistic regression. Many other medical scales used to assess severity of a patient have been developed using logistic regression. Logistic regression may be used to predict the risk of developing a given disease (e.g. diabetes; coronary heart disease), based on observed

characteristics of the patient (age, sex, body mass index, results of various blood tests, etc.).Another example might be to predict whether a Nepalese voter will vote Nepali Congress or Communist Party of Nepal or Any Other Party, based on age, income, sex, race, state of residence, votes in previous elections, etc. The technique can also be used in engineering, especially for predicting the probability of failure of a given process, system or product. It is also used in marketing applications such as prediction of a customer's propensity to purchase a product or halt a subscription, etc. In economics it can be used to predict the likelihood of a person ending up in the labor force, and a business application would be to predict the likelihood of a homeowner defaulting on a mortgage. Conditional random fields, an extension of logistic regression to sequential data, are used in natural language processing.

**Logistic Regression in Machine Learning**

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. 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**.
- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
- 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:



**Logistic Function (Sigmoid Function):**

- The sigmoid function is a mathematical function used to map the predicted values to probabilities.
- It maps any real value into another value within a range of 0 and 1.
- The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.
- In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

**Assumptions for Logistic Regression:**

- The dependent variable must be categorical in nature.
- The independent variable should not have multi-collinearity.

**Logistic Regression Equation:**

The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:

- We know the equation of the straight line can be written as:

$$y= b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

- In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):

$$\frac{y}{1-y}\text{; 0 for y= 0, and infinity for y=1}$$

- But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

$$log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

The above equation is the final equation for Logistic Regression.

**Type of Logistic Regression:**

On the basis of the categories, Logistic Regression can be classified into three types:

- **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

- **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
- **Ordinal:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

**Steps in Logistic Regression:**

To implement the Logistic Regression using Python, we will use the same steps as we have done in previous topics of Regression. Below are the steps:

- Data Pre-processing step
- Fitting Logistic Regression to the Training set
- Predicting the test result
- Test accuracy of the result(Creation of Confusion matrix)
- Visualizing the test set result.

**Python Code-**

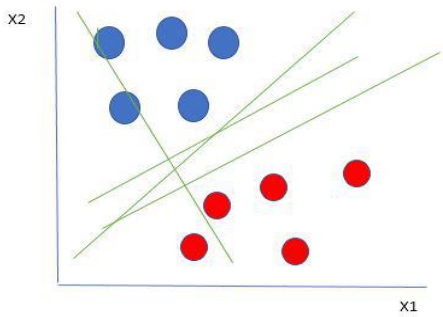
**Logistic Regression Classifier**

```
In [45]: from sklearn.linear_model import LogisticRegression
         from sklearn.pipeline import Pipeline
         clf_logReg_pipe = Pipeline([("vect", CountVectorizer()), ("tfidf", TfidfTransformer()), ("clf_logReg", LogisticRegression())])
         clf_logReg_pipe.fit(X_train, X_train_targetSentiment)

         import numpy as np
         predictedLogReg = clf_logReg_pipe.predict(X_test)
         np.mean(predictedLogReg == X_test_targetSentiment)*100

Out[45]: 93.6326884204447
```

## 3.5.3. Support Vector Machine Classifier

Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

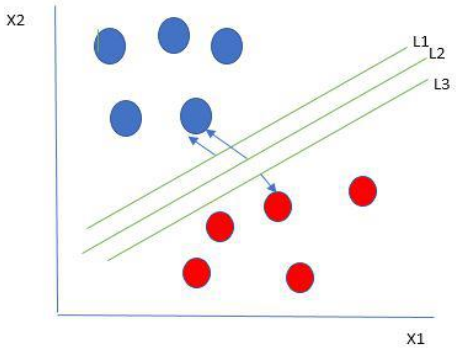
Let’s consider two independent variables x1, x2 and one dependent variable which is either a blue circle or a red circle.




*Linearly Separable Data points*

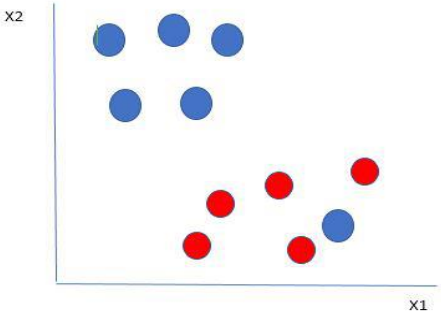
From the figure above its very clear that there are multiple lines (our hyperplane here is a line because we are considering only two input features x1, x2) that segregates our data points or does a classification between red and blue circles. So how do we choose the best line or in general the best hyperplane that segregates our data points.

**Selecting the best hyper-plane:**
One reasonable choice as the best hyperplane is the one that represents the largest separation or margin between the two classes.

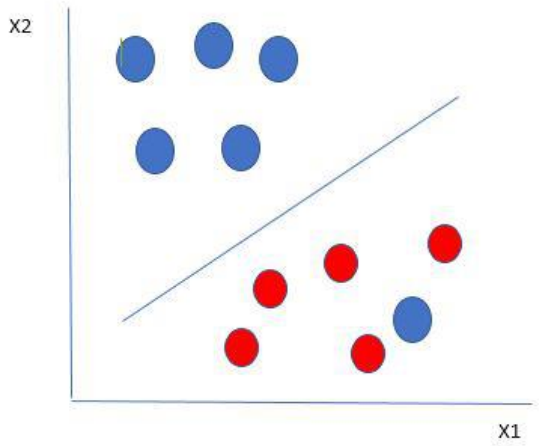



So we choose the hyperplane whose distance from it to the nearest data point on each side is maximized. If such a hyperplane exists it is known as the maximum-margin hyperplane/hard margin. So from the above figure, we choose L2.

Let's consider a scenario like shown below

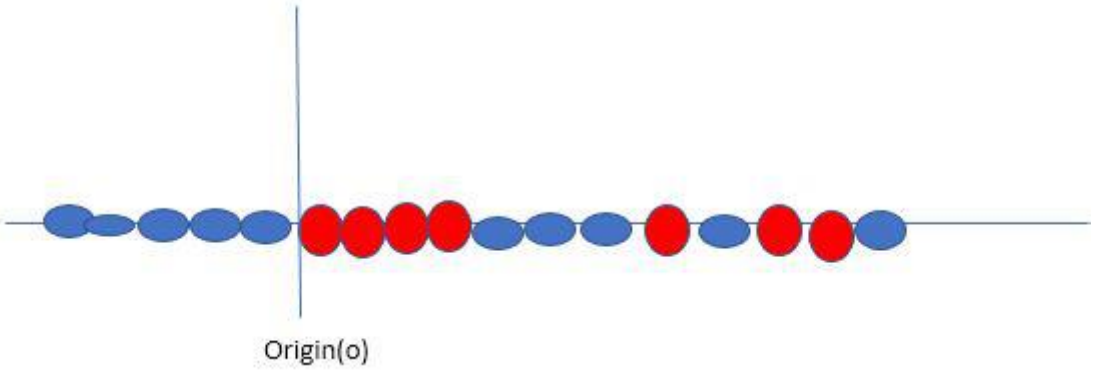


Here we have one blue ball in the boundary of the red ball. So how does SVM classify the data? It's simple! The blue ball in the boundary of red ones is an outlier of blue balls. The SVM algorithm has the characteristics to ignore the outlier and finds the best hyperplane that maximizes the margin. SVM is robust to outliers.

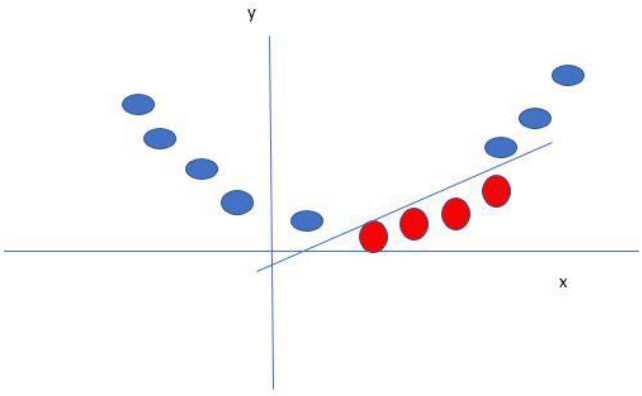



So in this type of data points what SVM does is, it finds maximum margin as done with previous data sets along with that it adds a penalty each time a point crosses the margin. So the margins in these type of cases are called soft margin. When there is a soft margin to the data set, the SVM tries to minimize *(1/margin+∧(∑penalty))*. Hinge loss is a commonly used penalty. If no violations no hinge loss.If violations hinge loss proportional to the distance of violation.

Till now, we were talking about linearly separable data(the group of blue balls and red balls are separable by a straight line/linear line). What to do if data are not linearly separable?




Say, our data is like shown in the figure above.SVM solves this by creating a new variable using a kernel. We call a point $x_i$ on the line and we create a new variable $y_i$ as a function of distance from origin o.so if we plot this we get something like as shown below




In this case, the new variable y is created as a function of distance from the origin. A non-linear function that creates a new variable is referred to as kernel.

**SVM Kernel:**
The SVM kernel is a function that takes low dimensional input space and transforms it into higher-dimensional space, ie it converts not separable problem to separable problem. It is mostly useful in non-linear separation problems. Simply put the kernel, it does some extremely complex data transformations then finds out the process to separate the data based on the labels or outputs defined.

**Advantages of SVM:**

- Effective in high dimensional cases
- Its memory efficient as it uses a subset of training points in the decision function called support vectors
- Different kernel functions can be specified for the decision functions and its possible to specify custom kernels

**Python Code-**

**Support Vector Machine Classifier**

```
In [46]: from sklearn.svm import LinearSVC
         clf_linearSVC_pipe = Pipeline([("vect", CountVectorizer()), ("tfidf", TfidfTransformer()), ("clf_linearSVC", LinearSVC())])
         clf_linearSVC_pipe.fit(X_train, X_train_targetSentiment)

         predictedLinearSVC = clf_linearSVC_pipe.predict(X_test)
         np.mean(predictedLinearSVC == X_test_targetSentiment)*100

Out[46]: 93.60381172393878
```

## 3.5.4. Decision Tree Classifier

Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.**

In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed on the basis of features of the given dataset.

***It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.***

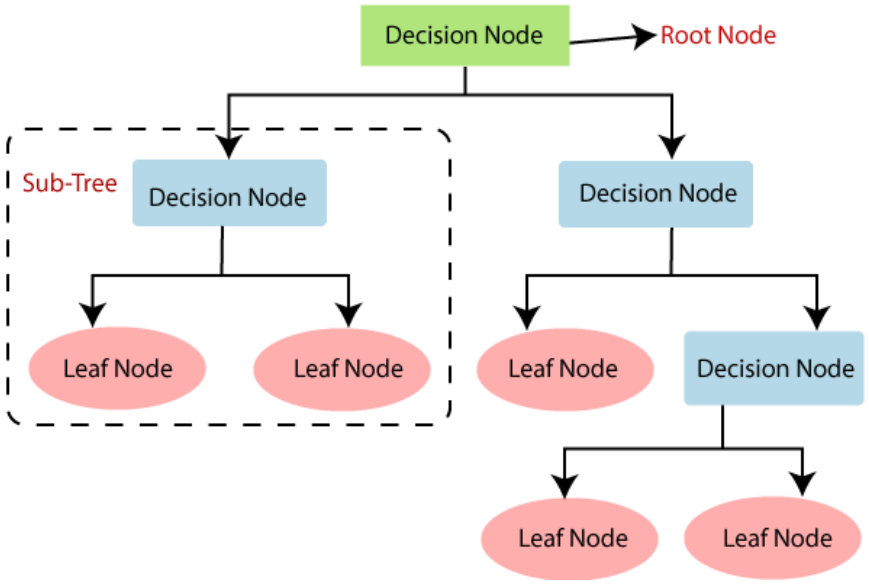
It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.

In order to build a tree, we use the **CART algorithm,** which stands for **Classification and Regression Tree algorithm.**

A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

Below diagram explains the general structure of a decision tree:

**Note: A decision tree can contain categorical data (YES/NO) as well as numeric data.**



**Why use Decision Trees?**

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.

The logic behind the decision tree can be easily understood because it shows a tree-like structure.

**Decision Tree Terminologies**

 **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.

 **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.

 **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.

 **Branch/Sub Tree:** A tree formed by splitting the tree.

 **Pruning:** Pruning is the process of removing the unwanted branches from the tree.

 **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

**How does the Decision Tree algorithm Work?**

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

**Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.

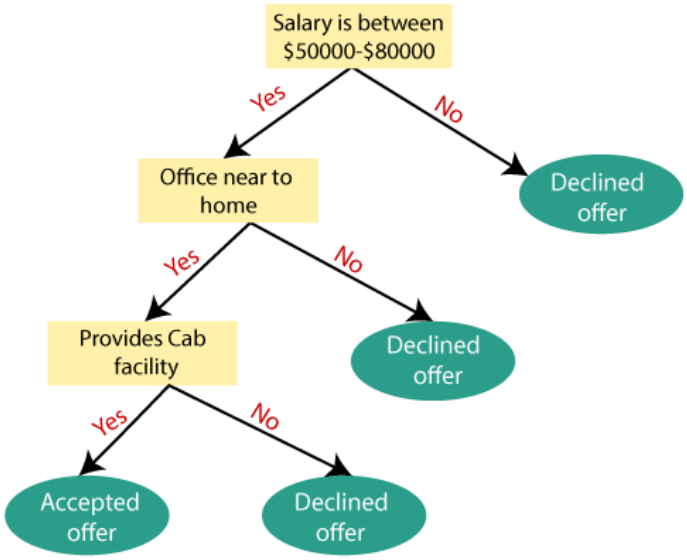
**Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**

**Step-3:** Divide the S into subsets that contains possible values for the best attributes.

**Step-4:** Generate the decision tree node, which contains the best attribute.

**Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

**Example:** Suppose there is a candidate who has a job offer and wants to decide whether he should accept the offer or Not. So, to solve this problem, the decision tree starts with the root node (Salary attribute by ASM). The root node splits further into the next decision node (distance from the office) and one leaf node based on the corresponding labels. The next decision node further gets split into one decision node (Cab facility) and one leaf node. Finally, the decision node splits into two leaf nodes (Accepted offers and Declined offer). Consider the below diagram:




**Attribute Selection Measures**

While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as **Attribute selection measure or ASM.** By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:

**Information Gain**

**Gini Index**

**1. Information Gain:**

Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.

It calculates how much information a feature provides us about a class.

According to the value of information gain, we split the node and build the decision tree.

A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below formula:

Information Gain= Entropy(S)- [(Weighted Avg) *Entropy(each feature)

**Entropy:** Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as:

Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)

**Where,**

**S= Total number of samples**

**P(yes)= probability of yes**

**P(no)= probability of no**

**2. Gini Index:**

Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm.

An attribute with the low Gini index should be preferred as compared to the high Gini index.

It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.

Gini index can be calculated using the below formula:

Gini Index= $1- \sum_j P_j^2$

**Pruning: Getting an Optimal Decision tree**

*Pruning is a process of deleting the unnecessary nodes from a tree in order to get the optimal decision tree.*

A too-large tree increases the risk of overfitting, and a small tree may not capture all the important features of the dataset. Therefore, a technique that decreases the size of the learning tree without reducing accuracy is known as Pruning. There are mainly two types of tree **pruning** technology used:

**Cost Complexity Pruning**

**Reduced Error Pruning.**

### Advantages of the Decision Tree

It is simple to understand as it follows the same process which a human follow while making any decision in real-life.

It can be very useful for solving decision-related problems.

It helps to think about all the possible outcomes for a problem.

There is less requirement of data cleaning compared to other algorithms.

### Disadvantages of the Decision Tree

The decision tree contains lots of layers, which makes it complex.

It may have an overfitting issue, which can be resolved using the **Random Forest algorithm.**

For more class labels, the computational complexity of the decision tree may increase.

**Python Code-**

## Decision Tree Classifier

In [48]:
```
from sklearn.tree import DecisionTreeClassifier
clf_decisionTree_pipe = Pipeline([("vect", CountVectorizer()), ("tfidf", TfidfTransformer()),
                                  ("clf_decisionTree", DecisionTreeClassifier())])
clf_decisionTree_pipe.fit(X_train, X_train_targetSentiment)

predictedDecisionTree = clf_decisionTree_pipe.predict(X_test)
np.mean(predictedDecisionTree == X_test_targetSentiment)*100
```

Out[48]:
```
89.84984117816921
```

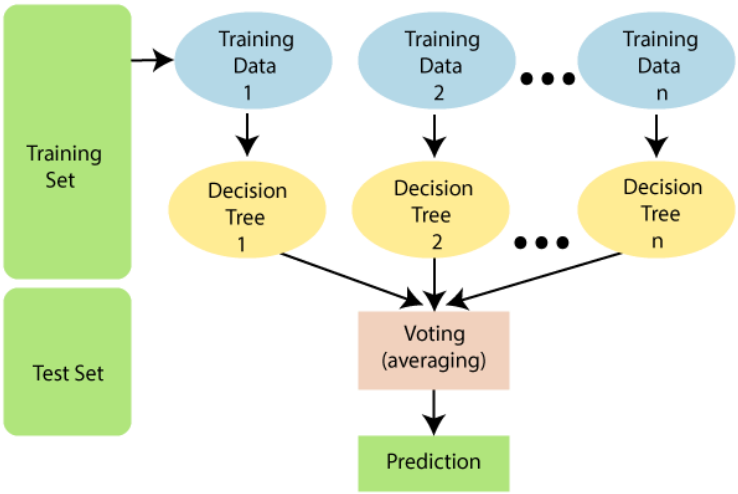
## 3.5.5. Random Forest (Ensemble Learning)

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model.*

As the name suggests, ***"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."*** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

**The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.**

The below diagram explains the working of the Random Forest algorithm:




### Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

## Working of Random Forest

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

**Step-1:** Select random K data points from the training set.

**Step-2:** Build the decision trees associated with the selected data points .

**Step-3:** Choose the number N for decision trees that you want to build.

**Step-4:** Repeat Step 1 & 2.

**Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

The working of the algorithm can be better understood by the below example:

**Example:** Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision. Consider the below image:

## Applications of Random Forest

There are mainly four sectors where Random forest mostly used:

1. **Banking:** Banking sector mostly uses this algorithm for the identification of loan risk.
2. **Medicine:** With the help of this algorithm, disease trends and risks of the disease can be identified.
3. **Land Use:** We can identify the areas of similar land use by this algorithm.
4. **Marketing:** Marketing trends can be identified using this algorithm.

## Advantages of Random Forest

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

## Python Implementation of Random Forest Algorithm

Now we will implement the Random Forest Algorithm tree using Python. For this, we will use the same dataset "user_data.csv", which we have used in previous classification models. By using the same dataset, we can compare the Random Forest classifier with other classification models such as Decision tree Classifier,KNN,SVM ,Logistic Regression etc.

Implementation Steps are given below:

- Data Pre-processing step
- Fitting the Random forest algorithm to the Training set
- Predicting the test result
- Test accuracy of the result (Creation of Confusion matrix)
- Visualizing the test set result.

## Important Hyperparameters

Hyperparameters are used in random forests to either enhance the performance and predictive power of models or to make the model faster.

*Following hyperparameters increases the predictive power:*

1. **n_estimators**– number of trees the algorithm builds before averaging the predictions.
2. ***max_features***– maximum number of features random forest considers splitting a node.
3. ***mini_sample_leaf***– determines the minimum number of leaves required to split an internal node.

***Following hyperparameters increases the speed:***

1. ***n_jobs***– it tells the engine how many processors it is allowed to use. If the value is 1, it can use only one processor but if the value is -1 there is no limit.
2. ***random_state***– controls randomness of the sample. The model will always produce the same results if it has a definite value of random state and if it has been given the same hyperparameters and the same training data.
3. ***oob_score*** – *OOB* means out of the bag. It is a random forest cross-validation method. In this one-third of the sample is not used to train the data instead used to evaluate its performance. These samples are called out of bag samples.

## Python Code-

**Ensemble classifier (Random forest)**

```
In [49]: from sklearn.ensemble import RandomForestClassifier
         clf_ess_pipe = Pipeline([("vect", CountVectorizer()), ("tfidf", TfidfTransformer()), ("clf_ess", RandomForestClassifier())])
         clf_ess_pipe.fit(X_train, X_train_targetSentiment)
         predictedess = clf_ess_pipe.predict(X_test)
         np.mean(predictedess == X_test_targetSentiment)*100

Out[49]: 93.35835980363846
```

# Chapter-4

## 4.1. Result Analysis

Looks like all the models performed very well (>90%), and we will use the Support Vector Machine Classifier since it has the highest accuracy level at 93.94%.

The model was evaluated using a confusion matrix. This is a contingency table of predicted and actual labels which shows how the model performed in predicting the polarity of the test reviews. The metrics for that include accuracy, Kappa, sensitivity, and specificity.

Accuracy is a metric that measures the fraction of predictions our model got right. It can be calculated using the following formula:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

Accuracy can also be expressed as true positives and negatives, and false positives and negatives.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Kappa statistic is a measure of how close the classified data is to the ground truth labels. It is calculated using the following formula:

$$Kappa = \frac{Total\ accuracy - Random\ accuracy}{1 - Random\ accuracy}$$

Sensitivity is the rate of the model in correctly classifying the data of class 1 (the positive class). The formula used to calculate sensitivity as follows:

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

Specificity is the rate of the model correctly classifying the data of class 0 (the negative class). The formula used to calculate specificity as follows:

$$Specificity = \frac{TN}{TN + FP}$$

```
In [52]: from sklearn.metrics import classification_report
         from sklearn.metrics import accuracy_score

         print(classification_report(X_test_targetSentiment, predictedGS_clf_LinearSVC_pipe))
         print('Accuracy: {}'. format(accuracy_score(X_test_targetSentiment, predictedGS_clf_LinearSVC_pipe)))
```

```
              precision    recall  f1-score   support

                   0.00      0.00      0.00         3
    Negative       0.59      0.22      0.32       165
     Neutral       0.35      0.06      0.10       307
    Positive       0.94      0.99      0.97      6451

    accuracy                           0.93      6926
   macro avg       0.47      0.32      0.35      6926
weighted avg       0.91      0.93      0.91      6926

Accuracy: 0.9343055154490326
```

Accuracy: 0.9408027721628646 Below is the summary of the classification report:

- Precision: determines how many objects selected were correct
- Recall: tells you how many of the objects that should have been selected were actually selected
- F1 score measures the weights of recall and precision (1 means precision and recall are equally important, 0 otherwise)
- Support is the number of occurrences of each class

The results in this analysis confirms our previous data exploration analysis, where the data are very skewed to the positive reviews as shown by the lower support counts in the classification report. Also, both neutral and negative reviews has large standard deviation with small frequencies, which we would not consider significant as shown by the lower precision, recall and F1 scores in the classification report.

However, despite that Neutral and Negative results are not very strong predictors in this data set, it still shows a 94.08% accuracy level in predicting the sentiment analysis, which we tested and worked very well when inputting arbitrary text (new_text). Therefore, we are comfortable here with the skewed data set. Also, as we continue to input new dataset in the future that is more balanced, this model will then re-adjust to a more balanced classifier which will increase the accuracy level.

Note: The first row will be ignored as we previously replaced all NAN with " ". We tried to remove this row when we first imported the raw data, but Pandas DataFrame did not like this row removed when we tried to drop all NAN (before stratifying and splitting the dataset). As a result, replacing the NAN with " " was the best workaround and the first row will be ignored in this analysis.

Finally, the overall result here explains that the products in this dataset are generally positively rated.

```
In [53]: from sklearn import metrics
         metrics.confusion_matrix(X_test_targetSentiment, predictedGS_clf_LinearSVC_pipe)

Out[53]: array([[   0,    0,    0,    3],
                [   0,   36,    8,  121],
                [   0,   17,   18,  272],
                [   0,    8,   26, 6417]], dtype=int64)
```

Note: The first row and column will be ignored as we previously replaced all NAN with " ". This is the same situation explained above in the classification report.

By considering only row 2-4 and column 2-4 labeled as negative, neutral and positive, we see that positive sentiment can sometimes be confused for one another with neutral and negative ratings, with scores of 246 and 104 respectively. However, based on the overall number of significant positive sentiment at a score 6445, then confusion score of 246 and 104 for neutral and negative ratings respectively are considered insignificant.

Also, this is a result of positively skewed dataset, which is consistent with both our data exploration and sentiment analysis. Therefore, we conclude that the products in this dataset are generally positively rated, and should be kept from Amazon's product roster.

## Chapter -5

# 5.1. Risk Analysis

**Dealing with class-imbalance**

Class imbalance is an imbalance of class distribution of training. The ideal situation is that the number of positive and negative samples is similar. However, if there are 995 positive samples and only 5 negative samples, it means that there is class imbalance. From the perspective of training model, if the number of samples in a category is small, the "information" provided by this category is too small, then the model does not learn how to identify a few classes. In this paper, a classical oversampling algorithm called SMOTE (Synthetic Minority Oversampling) is adopted-the minority class is over-sampled by creating "synthetic" examples rather than substitution (Chawla et al., 2002).Then the confusion matrix has been changed into below.

Table4: The confusion matrix of the Naive Bayes with SMOTE

| | Predicted negative | Predicted positive |
|---|---|---|
| Actual negative | 85 | 38 |
| Actual positive | 90 | 732 |

Table5:The confusion matrix of the Logistic Regression with SMOTE

| | Predicted negative | Predicted positive |
|---|---|---|
| Actual negative | 91 | 32 |
| Actual positive | 123 | 699 |

Comparing the confusion matrix before and after SMOTE, it can be observed that the model prediction in negative has been improved. The Receiver Operating Characteristic (ROC) curve can summarize the performance of classifier and the Area under the Curve (AUC) is a traditional performance metric for a ROC curve.AUC is the area under the Roc curve, between 0.1 and 1. As a numerical value, AUC can directly evaluate the quality of classifier. The larger the value, the better the model.

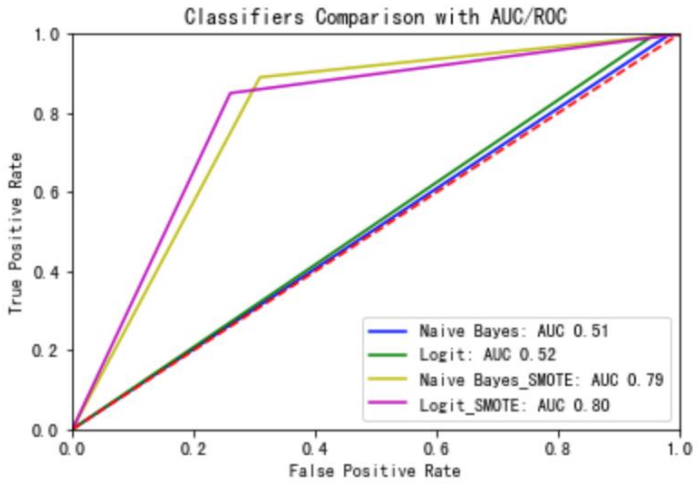



Figure10: Classifiers comparison

From the figure10, it can be seen that the AUC scores of the two models has improved from 0.5 to about 0.8 after applying the SMOTE, meaning the model turned to be better after the adjustment.

## Chapter -6

# Conclusion

In conclusion the analysis findings show from the customer reviews that language and words used to review products were likely to be used also by customers who review movies and game products from Amazon. The correlation analysis using the word frequency between the three products was statistically significant indicating that the frequency of words used in the three products was similar. Anticipation was one of the sentiments that was elicited from the customer reviews and most customers talked about time as most frequently, hence anticipation can be illustrated in customer reviews when they talk about timing of products. To be able to predict disgust, joy, fear, sadness, surprise, trust, positive and negative sentiments from customer reviews the most frequent words associated with these sentiments are taken into account. Another analysis that is essential in portraying the most positive and negative of words used is the Term frequency – inverse Document frequency analysis, this method plots the most frequent terms in both positive and negative sentiments hence providing a clear picture of the reviews data.

In the classification of reviews dataset into either positive or negative sentiment group, a Support Vector (SVC) classification model was used. The model performed quite well. The metrics used to measure the model performance were accuracy, Kappa, sensitivity and specificity. These metrics show that the model had an accuracy of 93.35% in classifying reviews and the accuracy similar to the sensitivity and specificity scores, but the Kappa score was quite low. This is worrisome because our model was quite ambitious, and the Kappa score shows there is still some improvement to do to ensure the model is in agreement with ground truth instead of just focusing on true positive and true negatives.

From the analysis above in the classification report, we can see that products with lower reviews are not significant enough to predict these lower rated products are inferior. On the other hand, products that are highly rated are considered superior products, which also perform well and should continue to sell at a high level.

As a result, we need to input more data in order to consider the significance of lower rated product, in order to determine which products should be dropped from Amazon's product roster.

The good news is that despite the skewed dataset, we were still able to build a robust Sentiment Analysis machine learning system to determine if the reviews are positive or negative. This is possible as the machine learning system was able to learn from all the positive, neutral and negative reviews, and fine tune the algorithm in order to avoid bias sentiments.

In conclusion, although we need more data to balance out the lower rated products to consider their significance, however we were still able to successfully associate positive, neutral and negative sentiments for each product in Amazon's Catalog.