UNIVERSITY of GREENWICH

Course Title: **Business Intelligence and Data Mining COMP1615**

Student Name: Mohamed Al-kaisi

Student ID : 000931504

Demo link : https://gre.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=52597bba-ec4e-4461-bd5a-aba50149a7a6

Contents

Table of Figures

Introduction

Online shopping industry increasing with technology advancement year by year. My choosing domain for BI application was the retail industry. Understanding how the marketing industry evaluates customer stratification and target customers with a specific product is an excellent skill to have. Over the years, it starts to become necessary to understand part of data gathering is what other people think. The constant growing popularity and availability of resources such as personal blogs and online reviews site and as most retail locations ask their own customers to review the product they bought on its own page. Companies do not have enough resources to view millions of reviews, but a handy and accessible data science approach can be used to identify feelings, or positive or negative reviews is called sentiment analysis or opinion mining. The method considered to be one of the most popular applications of text analytics. It also widely used technique and especially with social media analysis for any domain such as business or about move or a product or in our situation understating the reception by people from their own reviews.

Text data in unstructured format can be classified into two different types within sentiment analysis applications, such as factual based or opinion based subjective or a feature aspect based. The accurate based is referring to a text that has a subjective context such as data from social media, surveys, feedback form, and any opinionated and expresses the beliefs, judgment, and feelings of humans. The feature aspect is identifying opinion by assessing different factors of entity such as a product screen or a bank, and so on.

Amazon is known to be the largest E-commerce website and proven to have numerous amounts of reviews that can be seen. The dataset used for this project called Amazon Cell Phone Reviews posted in Kaggle by Nibras (2020). The dataset was unlabelled, and to use with supervised machine learning; the data needs to be labeled. The idea is converting the rating column into the positive column by classifying 1,2 into 0 as negative and 4, 5 as positive so that column can enable the data with supervised machine learning alongside the review text. The dataset consists of 67000 after merging both items and review files together using the “sin” column of cell phone reviews for different varieties of phones such as Samsung, one plus, Nokia, Motorola, HUAWEI, Google, Apple, Asus, Xiaomi and Sony.

1. Literature review

1.1 Introduction

The improvement of technology in this day of age has undertaken a big shift by reducing the need to visit a shop to buy specific a product. The idea of online shopping shifted with the improvement of technology, but some questions appeared on how the experience of shopping can be transferred online as most of the important customer influence is trust. Trust that the product details online fit in with an actual product, and that's where the idea of asking customers to write a review under the product after bought so for other customers can read it before making the purchase. Reviews help customers to understand more about the products by reading about customer experience, but also the technique also helps the company to improve its product and customer satisfaction. Amazon online shopping includes reviews under each product from customers, which shows all customers experience with the product before consuming.

According to Fuel (2016), 94% of online consumers read reviews before making any purchasing decisions. Spiegel Research Centre also published a pdf with the title How Online Reviews Influence Sales, (2017), stating that 95% of shoppers read reviews before making any purchase. Products online with no reviews or ratings can display potential distrust for distrusting the product. Positive reviews can predict future customers for deciding to purchase such a product, and similarly with negative reviews will often cause sales loss

Companies do not have time to read and understand each review for all their product, which is a problem for retail companies. However, with the advancement of technologies, a data mining application developed called sentiment analysis, which is an important part of text analytics. Using this machine learning tool helps to provide insight by automatically analyze product reviews and separating them into tags. Such positive, neutral, and negative.

The aim of this paper to identify the approaches used on the Amazon cell phone dataset that was posted on the Kaggle website here in Amazon Cell Phones Reviews, 2020). After identifying the approaches that have been taking on the same dataset or similar reviews by amazon, then identify our project approach. The aim more to explore more on the data and use different types of classification algorithms to check-in for the best accuracy result.

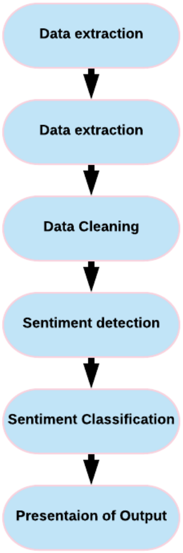


Figure 1 Sentiment Analysis Process

1.2 Related work

The first paper found in research that went over amazon reviews dataset for the goal to run business analytics and sentiment analysis was by Elli, M.S., and Wang, Y.F. (2016). They have stated that their use of tools demonstrated to be robust enough to give them high accuracy. Other than extracting sentiment and analyze the result also worked on detecting emotion from reviews, gender-based on the names, and detecting fake reviews. In terms of data mining algorithms, they ran classification algorithms such as multinomial Naïve Bayesian (MNB) and support vector machine (SVM) as the main classifier for their project. Another paper by Xu etc. (2014), working on similar data has applied existing supervised machine learning algorithms to predict reviews rating on a given numerical scale using only text. They split their data using 70% of the data for training and 30% of the data used for testing the model. The classifiers used on those data were used to determine precision and recall values such as Naïve Bayes and Multi-class SVM. The author Rain, C., 2013 also took the research to run sentiment analysis on Amazon product reviews for Kindle and used classification such as decision list and Naïve Bayesian as classifiers to tag positive or negative. The naive Bayes classifier performed better in their application than the decision list classifier with all their three-segmented dataset.

Bhatt, A, etc . (2015) aimed in the paper to build a system that visualizes the sentiment of the review as a form of charts. The author also applied classifiers such as Naive Bayes and SVM and maximum entropy to measure accuracy. Their result showed accurate enough for the test case used for iPhone 5 reviews.

Chen, W, etc. worked on Amazon Health & Personal Care product review using the rating mainly on review texts to find out what kind of words have positive and negative effects on rating using Bag-of-words. They have built three models, such as the latent factor model that considers only reviewers and product. Their second model adds on the interaction between words and other features, which is linear regression, and their last model was the support vector regression model to utilized unigrams and bigrams. Between both unigrams and bigrams, the unigrams produced for their work the most accurate results and are extremely useful for predicting rating for their larger variance. Unigrams provided for them higher performance, such as 15.89% bigger performance than bigrams.

In the paper by Shaikh, T., etc. (2016) used different techniques for extraction, selection for sentiment analysis. After their data collection, they have performed data pre-processing by removing stop words, special characters then applied phrase-level single word and multiword. The classifier used with this project was Naïve Bayes and concluded that this classifier gives a better result for phrase-level than a single word and multiword. A disadvantage of this paper, they have run only one classifier algorithm, which doesn't show enough results.

In the paper by Nasr, F.etc (2017) used several classification algorithms such as support vector machine (SVM), logistic regression, and decision result to predict the accuracy. The highest accuracy was giving using a support vector machine algorithm, which found that it doesn't work on a huge dataset well.

1.3 Methodology

The first setup of the project was downloading the data from Kaggle, then exploring the data to understand the variables given. The data comes with two files, such as items and reviews. The data will be merged based on asin column added in both data file which makes easy for analysis and segmentation. After the data exploring, the first result given was the dataset contains reviews for different cell phones sold by Amazon. The identified phone types were Samsung, one plus, Nokia, Motorola, HUAWEI, Google, Apple, Asus, Xiaomi, and Sony.

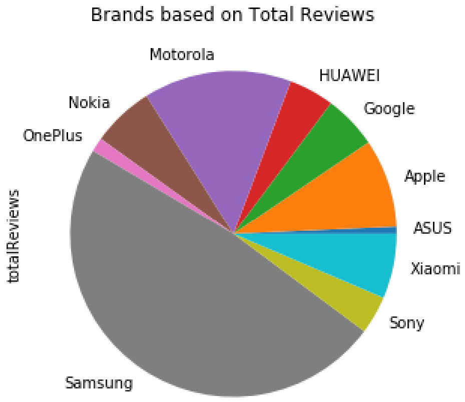


Figure 2 Pie chart showing brand based on total Reviews

Amazon website enables a rating from 1 to 5, which included in our dataset. The rating in the figure showing to have the highest amount of 5 stars rating and a 1-star rating.

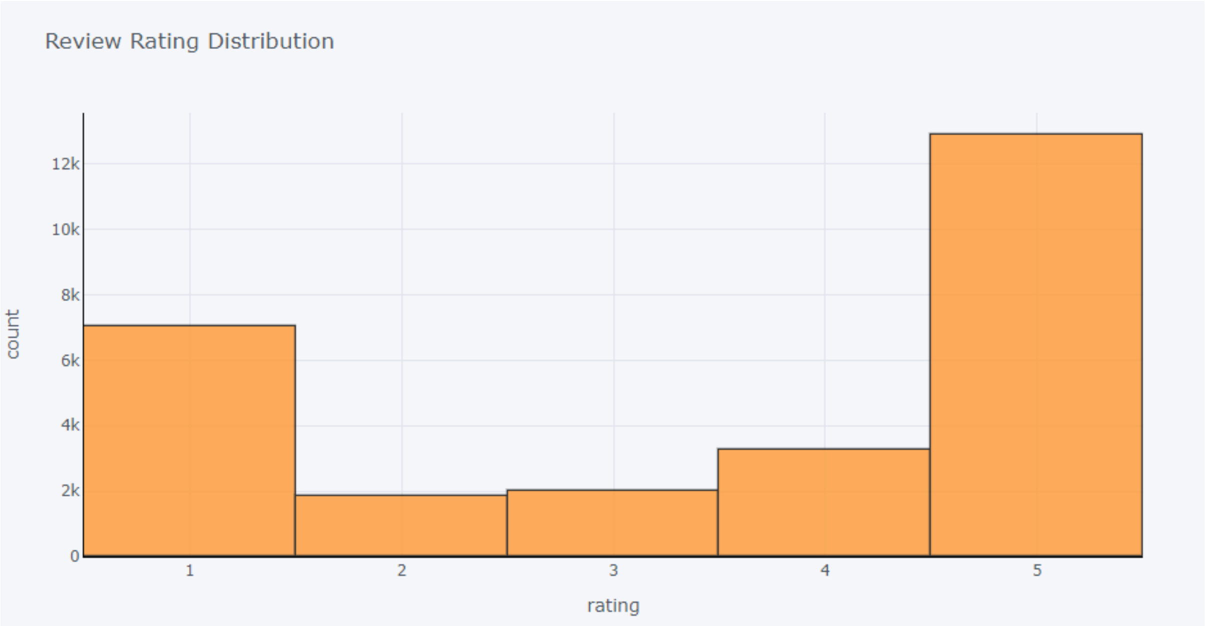


Figure 3 Rating total to understand what the common star rating is given

The rating will be used to create a positivity column classifying positive and negative reviews. Positive reviews are classified based on any rating that has either 4 or 5, and negative reviews are from rating on 1 or 2. Any reviews with rating 3 as natural will be dropped. Adding a positivity column will help in running machine learning classification as training data to perform supervised machine learning. Based on the rating used aggregation method to find out which phone has the most result, then use the bar chart to visualize the result below in figure 3.

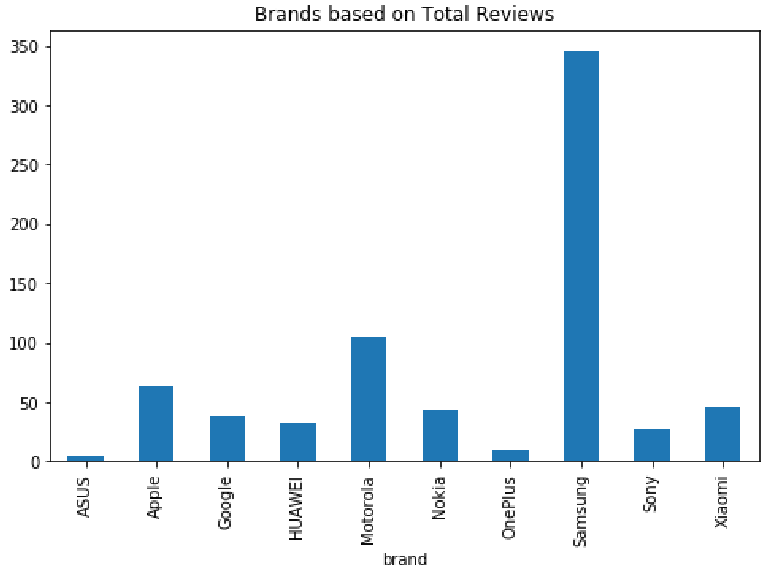


Figure 4 Brands based on Total Reviews

Figure 4 bar chart showing that Samsung, Motorola, and apple have the highest rating result, which similar results are showing in figure 2 in terms of total reviews. After converting based on ratings such as 1 and 2 classed as negative and 4 and 5 classed as positive then we run similar chart to count the brand in terms of positive and negative then the result similar to figure 3 that Samsung, Motorola, and Apple have the highest positive reviews based on the rating see figure 4 below.

From figure 2 to 3, The products will be analyzed with the highest total reviews, total rating, and most positive reviews based on the rating, which was Samsung, and Motorola.

1.4 Feature Extraction

Bag of Words used for our project to represent text data for the machine learning algorithm. It will be tokenized words for each and find out the frequency for each token. Example:

“Text Messaging Doesn't Work.”

“Text Messaging Work”

“Camera is the best.”

Each sentence treated as a separate document and made a list of all words from all the three documents without the punctuation.

“Text,” Message,” “doesn't,” “work,” camera,” work.”

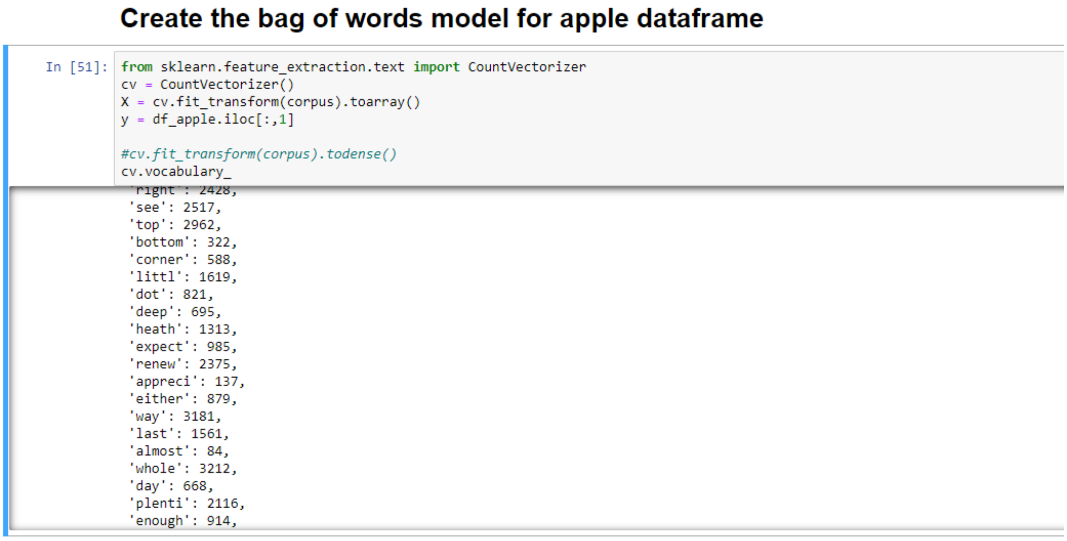
The next step for a bag of the word is to create a vector in order to be used by a machine learning algorithm. Vector created, for example, taking the first sentence, “Text Messaging doesn’t work,” we check the frequency of words from the ten unique words. The frequency of words would create a dictionary that contains all the words in our corpus as keys and the frequency of existence of the words as values.

“Text” = 1,”Message” = 1, “doesnt” = 1, “work” = 0,”camera” = 0 ,”work” = 0

The created document will be of “Text Messaging Doesn't Work” = [1,1,1,0,0,0]

In this approach bag of words, each word called is a gram, and creating a vocabulary of two-word pairs is called a bigram model. The process in here converting NLP text into numbers is called vectorization in machine learning.

After creating the dictionary, then time to create the bag of words model, which would create a matrix where columns correspond to the most frequent words in our dictionary where rows resemble the document. Example from our project creating a bag of words showing in figure 5.



Create the bag of words model for apple dataframe

```
In [51]: from sklearn.feature_extraction.text import CountVectorizer
         cv = CountVectorizer()
         X = cv.fit_transform(corpus).toarray()
         y = df_apple.iloc[:,1]

         #cv.fit_transform(corpus).todense()
         cv.vocabulary_
```

```
          'right': 2428,
          'see': 2517,
          'top': 2962,
          'bottom': 322,
          'corner': 588,
          'littl': 1619,
          'dot': 821,
          'deep': 695,
          'heath': 1313,
          'expect': 985,
          'renew': 2375,
          'appreci': 137,
          'either': 879,
          'way': 3181,
          'last': 1561,
          'almost': 84,
          'whole': 3212,
          'day': 668,
          'plenti': 2116,
          'enough': 914,
```

Figure 4 Bag of words example

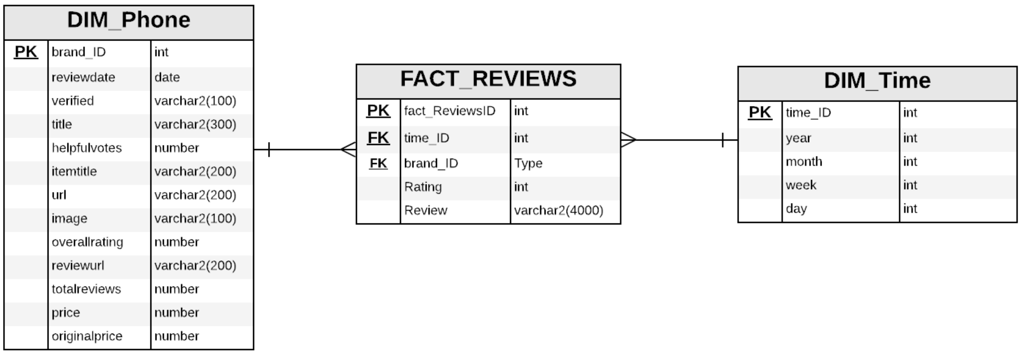
1.5 Classification Algorithm

The dataset in this section prepared for running classification algorithm as positive column created that represent positive and negative reviews and a bag of words method used on the text. Also, After researching other work on the dataset and understanding their choosing type of algorithm. Our methodology for classification that would be trying out Logistic Regression, Support Vector Machine, Kernal SVM, Naive Bayes, Decision Tree Classification, and finally, Random Forest. The highest accuracy given from two algorithms will be used for this project and explained after the implementation.

2. Data Collection from

Amazon knows to be the largest E-commerce website and proven to have in numerous amounts of reviews that can be seen. The dataset used for this project called Amazon Cell Phone Reviews posted in Kaggle by Nibras (2020). The dataset was unlabelled, and to use with supervised machine learning; the data needs to be labeled. The idea is converting the rating column into the positive column by classifying 1,2 into 0 as negative and 4, 5 as positive so that column can enable the data with supervised machine learning alongside the review text. The dataset consists of 67000 after merging both items and review files together using the "sin" column of cell phone reviews for different varieties of phones such as Samsung, one plus, Nokia, Motorola, HUAWEI, Google, Apple, Asus, Xiaomi and Sony. The data is worth the analysis to understand what particular phone actually bought and favored the most by customers from Amazon company.

3. An Entity Relationship Diagram (ERD) of the conceptual design and the relational schema developed to store the data in a data warehouse schema:



DIM_Phone		
PK	brand_ID	int
	reviewdate	date
	verified	varchar2(100)
	title	varchar2(300)
	helpfulvotes	number
	itemtitle	varchar2(200)
	url	varchar2(200)
	image	varchar2(100)
	overallrating	number
	reviewurl	varchar2(200)
	totalreviews	number
	price	number
	originalprice	number

FACT_REVIEWS		
PK	fact_ReviewsID	int
FK	time_ID	int
FK	brand_ID	Type
	Rating	int
	Review	varchar2(4000)

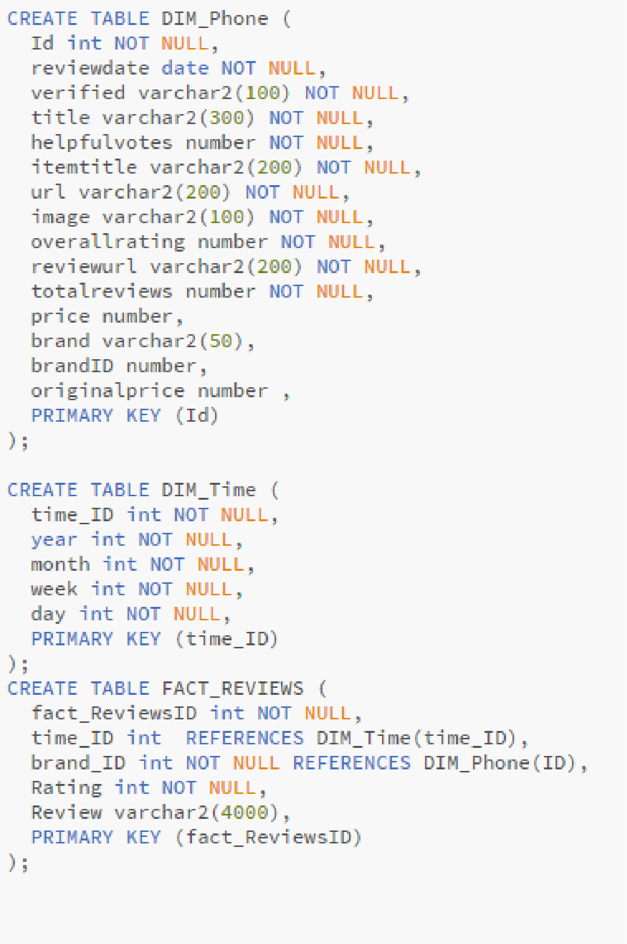
DIM_Time		
PK	time_ID	int
	year	int
	month	int
	week	int
	day	int

Figure 5 DW ERD

Link to ERD: https://www.lucidchart.com/invitations/accept/b620e56a-88dd-4159-ad87-95bc2de47374

Looking at the dataset using for this project, the main columns used for sentiment analysis is Reviews and rating for each phone brand. Both Rating and Reviews will be the main fact table with foreign keys connected to other tables. Dimensional table display time details when the review has been posted by users. Year, month, week, and day columns extracted from the date column given. The last table DIM phone display all the details of the phone, review details, and the name of the person who reviewed the phone. The goal for creating a data warehouse for our project was to explore the data such as finding out the brand that has the highest rating or understanding rating and reviews that created over a specific month and so on. However, for our project data warehouse isn't useful as I am using text and rating only columns.

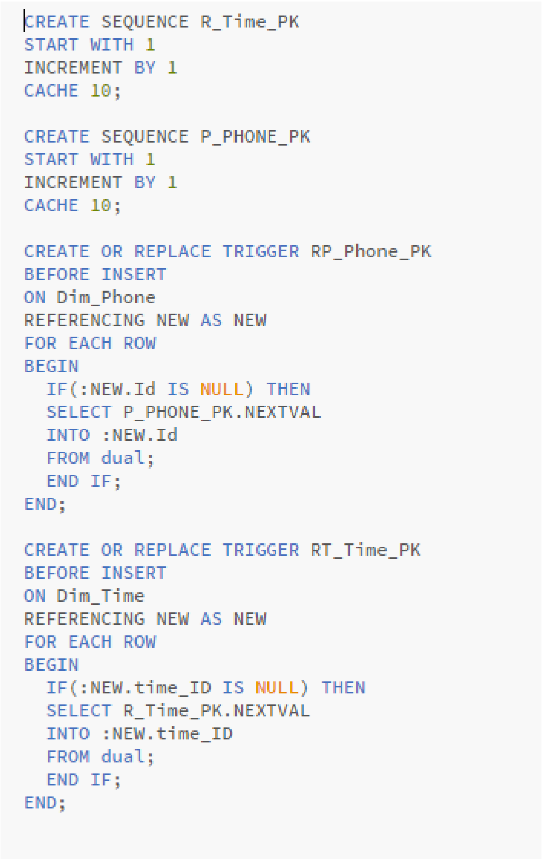
Figure 6 on the page below showing data warehouse schema code. The code includes forging and primary keys.



```
CREATE TABLE DIM_Phone (
  Id int NOT NULL,
  reviewdate date NOT NULL,
  verified varchar2(100) NOT NULL,
  title varchar2(300) NOT NULL,
  helpfulvotes number NOT NULL,
  itemtitle varchar2(200) NOT NULL,
  url varchar2(200) NOT NULL,
  image varchar2(100) NOT NULL,
  overallrating number NOT NULL,
  reviewurl varchar2(200) NOT NULL,
  totalreviews number NOT NULL,
  price number,
  brand varchar2(50),
  brandID number,
  originalprice number ,
  PRIMARY KEY (Id)
);

CREATE TABLE DIM_Time (
  time_ID int NOT NULL,
  year int NOT NULL,
  month int NOT NULL,
  week int NOT NULL,
  day int NOT NULL,
  PRIMARY KEY (time_ID)
);
CREATE TABLE FACT_REVIEWS (
  fact_ReviewsID int NOT NULL,
  time_ID int  REFERENCES DIM_Time(time_ID),
  brand_ID int NOT NULL REFERENCES DIM_Phone(ID),
  Rating int NOT NULL,
  Review varchar2(4000),
  PRIMARY KEY (fact_ReviewsID)
);
```

Figure 6 DW Schema

Primary keys auto-increment with Oracle is not as easy as using MYSQL. I had to created a sequence and trigger for each DIM_Phone and DIM_Time to make them auto increment ID. Figure 6 showing the code for triggers and schema.



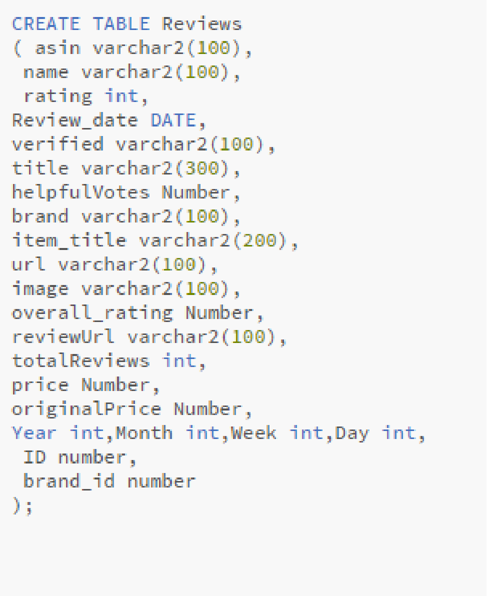
```
CREATE SEQUENCE R_Time_PK
START WITH 1
INCREMENT BY 1
CACHE 10;

CREATE SEQUENCE P_PHONE_PK
START WITH 1
INCREMENT BY 1
CACHE 10;

CREATE OR REPLACE TRIGGER RP_Phone_PK
BEFORE INSERT
ON Dim_Phone
REFERENCING NEW AS NEW
FOR EACH ROW
BEGIN
  IF(:NEW.Id IS NULL) THEN
  SELECT P_PHONE_PK.NEXTVAL
  INTO :NEW.Id
  FROM dual;
  END IF;
END;

CREATE OR REPLACE TRIGGER RT_Time_PK
BEFORE INSERT
ON Dim_Time
REFERENCING NEW AS NEW
FOR EACH ROW
BEGIN
  IF(:NEW.time_ID IS NULL) THEN
  SELECT R_Time_PK.NEXTVAL
  INTO :NEW.time_ID
  FROM dual;
  END IF;
END;
```

Figure 7 Triggers and Sequence created for primary to auto in oracle

3.1 The issue with using the university system database

Difficulties appeared with loading all my dataset after cleaning that contains 27000 rows. In Oracle tablespace, quotes exceeded the size given. I contacted the faculty IT staff to add extra tablespace quotes, and they did. Afterward, I was able only to load 200 rows of data into my data warehouse.

3.2 Create a staging table



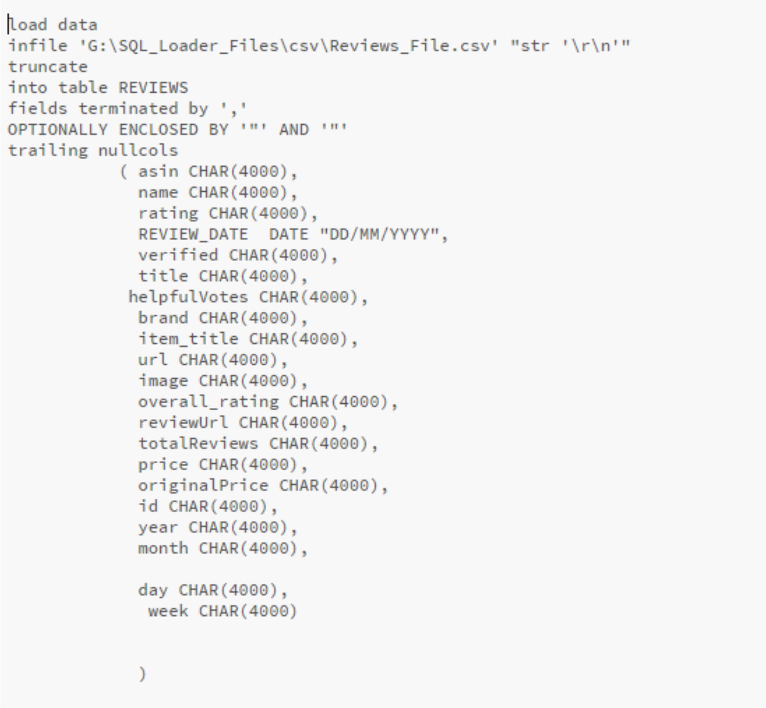
```
CREATE TABLE Reviews
( asin varchar2(100),
 name varchar2(100),
 rating int,
Review_date DATE,
verified varchar2(100),
title varchar2(300),
helpfulVotes Number,
brand varchar2(100),
item_title varchar2(200),
url varchar2(100),
image varchar2(100),
overall_rating Number,
reviewUrl varchar2(100),
totalReviews int,
price Number,
originalPrice Number,
Year int,Month int,Week int,Day int,
 ID number,
 brand_id number
);
```

Figure 8 DW staging table

Figure 8 showing the stage table code created based on the dataset CSV file columns. Creating a staging area will help to load the original CSV file dataset into the staging area table.

3.3 ETL solution

There are many tools available to perform ETL of data, such as Amazon's AWS Glue or Microsoft's SQL Server Integrated Services (SSIS). Since the work is being hosted in an Oracle SQL Developer database, it makes the most sense to use an Oracle ETL tool to import the data. Therefore, the ETL of data in this work is carried out by Oracle's SQL Loader tool.

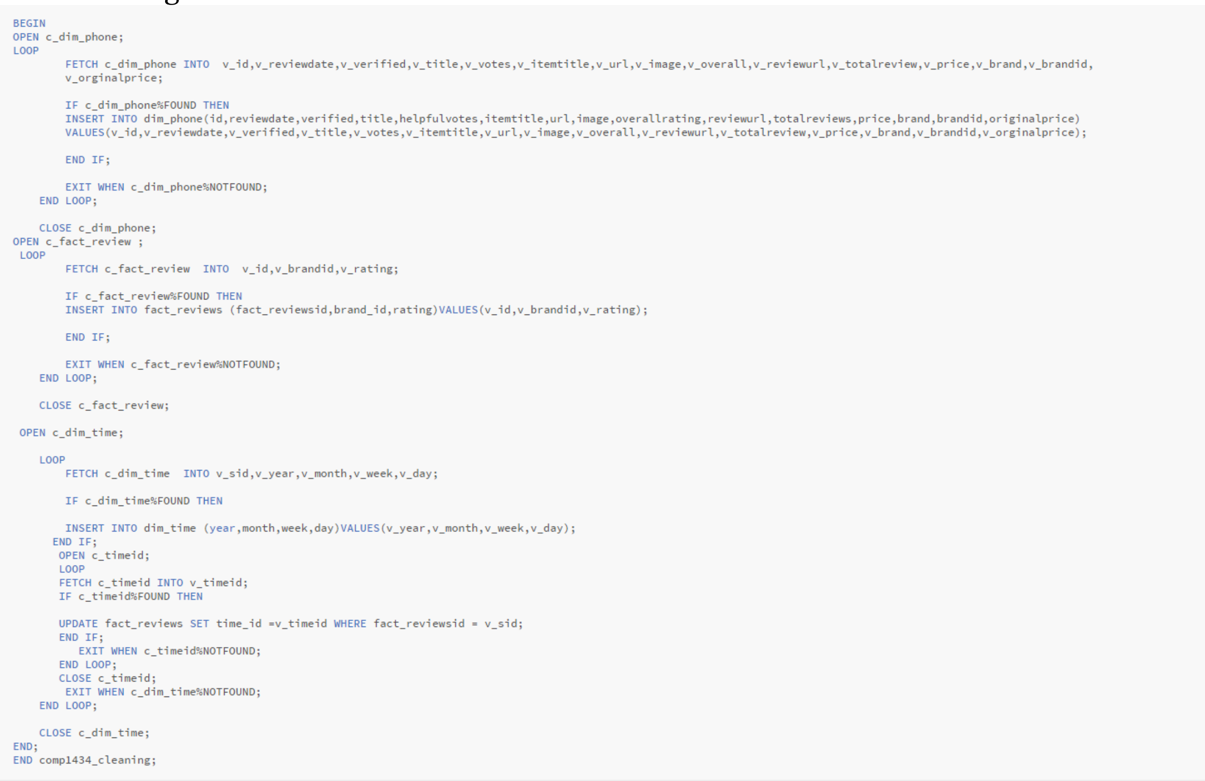


```
load data
infile 'G:\SQL_Loader_Files\csv\Reviews_File.csv' "str '\r\n'"
truncate
into table REVIEWS
fields terminated by ','
OPTIONALLY ENCLOSED BY '"' AND '"'
trailing nullcols
           ( asin CHAR(4000),
             name CHAR(4000),
             rating CHAR(4000),
             REVIEW_DATE  DATE "DD/MM/YYYY",
             verified CHAR(4000),
             title CHAR(4000),
            helpfulVotes CHAR(4000),
             brand CHAR(4000),
             item_title CHAR(4000),
             url CHAR(4000),
             image CHAR(4000),
             overall_rating CHAR(4000),
             reviewUrl CHAR(4000),
             totalReviews CHAR(4000),
             price CHAR(4000),
             originalPrice CHAR(4000),
             id CHAR(4000),
             year CHAR(4000),
             month CHAR(4000),

             day CHAR(4000),
              week CHAR(4000)


             )
```

Figure 9 SQL LOADER CTL file

3.4 PL/SQL procedure to move data into Datawarehouse

PL/SQL procedure created to move the ready data from the staging area into the data warehousing schema.

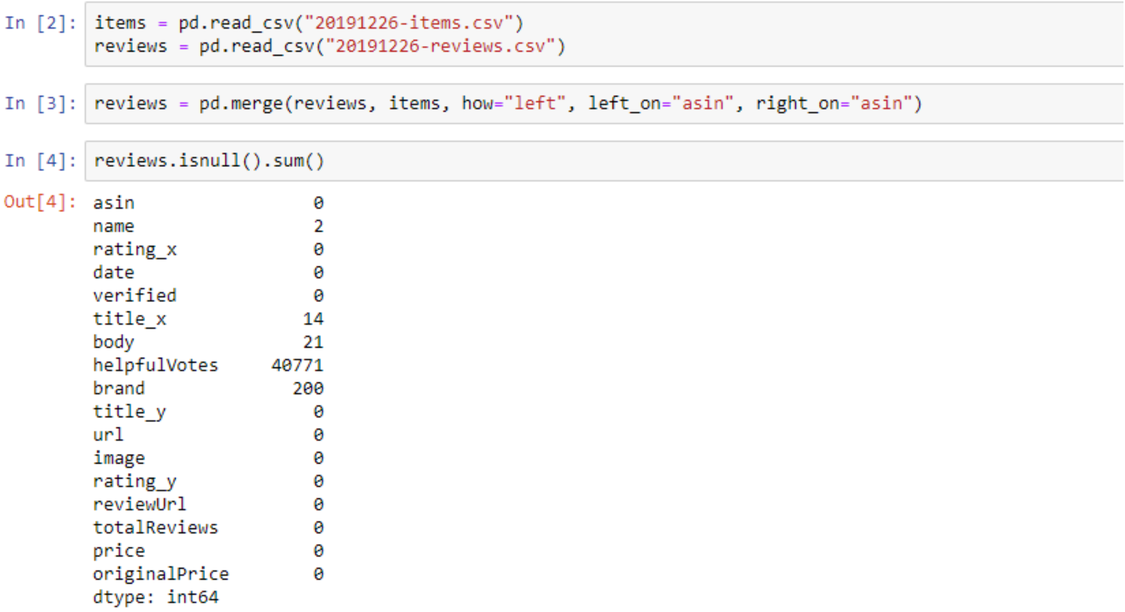


```
BEGIN
OPEN c_dim_phone;
LOOP
        FETCH c_dim_phone INTO  v_id,v_reviewdate,v_verified,v_title,v_votes,v_itemtitle,v_url,v_image,v_overall,v_reviewurl,v_totalreview,v_price,v_brand,v_brandid,
        v_orginalprice;

        IF c_dim_phone%FOUND THEN
        INSERT INTO dim_phone(id,reviewdate,verified,title,helpfulvotes,itemtitle,url,image,overallrating,reviewurl,totalreviews,price,brand,brandid,originalprice)
        VALUES(v_id,v_reviewdate,v_verified,v_title,v_votes,v_itemtitle,v_url,v_image,v_overall,v_reviewurl,v_totalreview,v_price,v_brand,v_brandid,v_orginalprice);

        END IF;

        EXIT WHEN c_dim_phone%NOTFOUND;
    END LOOP;

    CLOSE c_dim_phone;
OPEN c_fact_review ;
 LOOP
        FETCH c_fact_review  INTO  v_id,v_brandid,v_rating;

        IF c_fact_review%FOUND THEN
        INSERT INTO fact_reviews (fact_reviewsid,brand_id,rating)VALUES(v_id,v_brandid,v_rating);

        END IF;

        EXIT WHEN c_fact_review%NOTFOUND;
    END LOOP;

    CLOSE c_fact_review;

 OPEN c_dim_time;

    LOOP
        FETCH c_dim_time  INTO v_sid,v_year,v_month,v_week,v_day;

        IF c_dim_time%FOUND THEN

        INSERT INTO dim_time (year,month,week,day)VALUES(v_year,v_month,v_week,v_day);
      END IF;
       OPEN c_timeid;
       LOOP
       FETCH c_timeid INTO v_timeid;
       IF c_timeid%FOUND THEN

       UPDATE fact_reviews SET time_id =v_timeid WHERE fact_reviewsid = v_sid;
       END IF;
          EXIT WHEN c_timeid%NOTFOUND;
       END LOOP;
       CLOSE c_timeid;
        EXIT WHEN c_dim_time%NOTFOUND;
    END LOOP;

    CLOSE c_dim_time;
END;
END comp1434_cleaning;
```

Figure 10 PL/SQL Procedure to move data from staging into DW

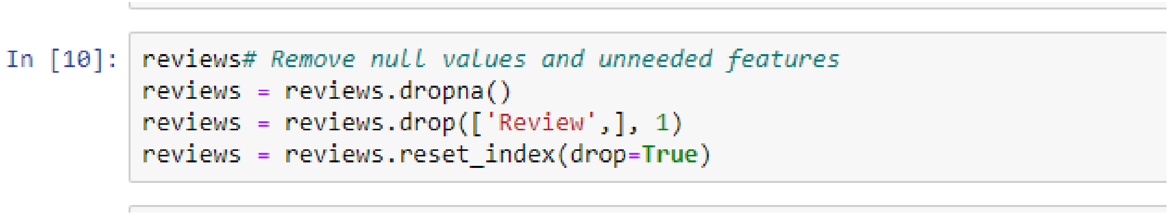
3.5 Data Cleaning

The first step into data cleaning was to merge both files together. It was based on merging identifying all the null values and removing them.



```
In [2]: items = pd.read_csv("20191226-items.csv")
        reviews = pd.read_csv("20191226-reviews.csv")

In [3]: reviews = pd.merge(reviews, items, how="left", left_on="asin", right_on="asin")

In [4]: reviews.isnull().sum()

Out[4]: asin                0
        name                2
        rating_x            0
        date                0
        verified            0
        title_x            14
        body               21
        helpfulVotes    40771
        brand             200
        title_y             0
        url                 0
        image               0
        rating_y            0
        reviewUrl           0
        totalReviews        0
        price               0
        originalPrice       0
        dtype: int64
```

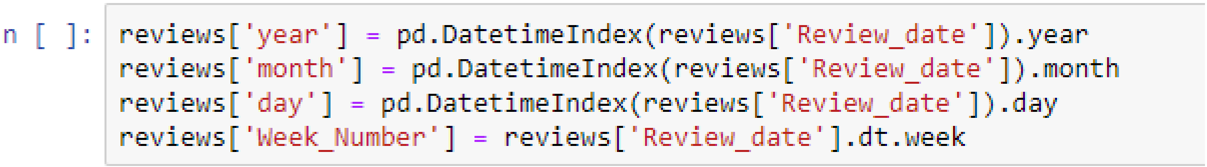
Figure 11 Files merged and null values deleted using the method in figure 12



```
In [10]: reviews# Remove null values and unneeded features
         reviews = reviews.dropna()
         reviews = reviews.drop(['Review',], 1)
         reviews = reviews.reset_index(drop=True)
```

Figure 12 method to drop all the null values

The final step was to prepare the columns needed for Time dimensional table in python as it is easier. The column extracted from the date column was the year, month, week, and day using the method in figure 13.



```
n [ ]: reviews['year'] = pd.DatetimeIndex(reviews['Review_date']).year
       reviews['month'] = pd.DatetimeIndex(reviews['Review_date']).month
       reviews['day'] = pd.DatetimeIndex(reviews['Review_date']).day
       reviews['Week_Number'] = reviews['Review_date'].dt.week
```

Figure 13 Method for extracting rows from date need for time dimensional table

4. Description and motivation of your data mining approach

The retail industry was the selection for this data mining project out of many others giving in the requirements. When we think about data mining projects created for businesses such as the retail industry, it always involves numbers. One part of the retail industry that improves with technology advancement is online shopping. The difficulties of this type of shopping are how the industry would evaluate customers' stratifications. Online shopping's have added loads of tools that can enhance customer relationship with the company to improve its stratification. One of the tools is allowing customers to rate and write reviews under the product page. Reviews enable the company to respond and help their own customers if there were any issues but also helps the company to attract more customers as reviews enable customers to trust the product is worthy of buying. Based on the reviews, the company improves its own products in order to attract more customers. However, the company would have to spend a long time to evaluate each product reviews. Therefore, using sentiment analysis approach can help to classify large data into positive, neutral, and negative then visualize them into a word cloud and other charts. Obviously, using the word cloud helps makes reading the reviews easier to understand how to enhance their services and products. After running sentiment analysis, approaches like cleaning the text and finding the most frequent words and other techniques.

However, instead of relying on manually crafted rules such as rule-based systems, text classification with machine learning approach learns to make a classification based on past observations by using the pre-labeled data for training data in which helps the machine learning algorithm to learn different associations between pieces of text and the particular output expected for a peritubular input. It also proves that the machine learning algorithm predicate more accurate results than a rule-based system. Before running, an algorithm needs to run feature extraction, which turns the text into a numerical representation in the form of vector. In our application used the approach bag of words, which makes a vector to represent the frequency of a word in a predefined dictionary of words.

Decision Tree classification is one of the algorithms used in our application. This algorithm classifies data into different classes by recursively separating the feature space into two parts and assigning different classes based on which region in the divided space a sentence is or based on its features. Support Vector machine is another method that follows a statistical classification approach based on the maximization of the margin between the instances and the separating hyper-plane. Many of the literature reviews read and cited in section 1 consider the algorithm support vector machine as the best text classification method. Naive Bayes method used in this project but did not give a higher result but its popular method in text categorization because of its simplicity and efficiency. In our application, we used logistic regression, which gave us a result of accuracy over 80%. Logistic regression is a statistical method for analyzing data set in which are one or more independent variables that determine an outcome.

5. Design and Implementation

5.1 Design

Figure 11 Describing how the project will be tackled to achieve our main objectives.

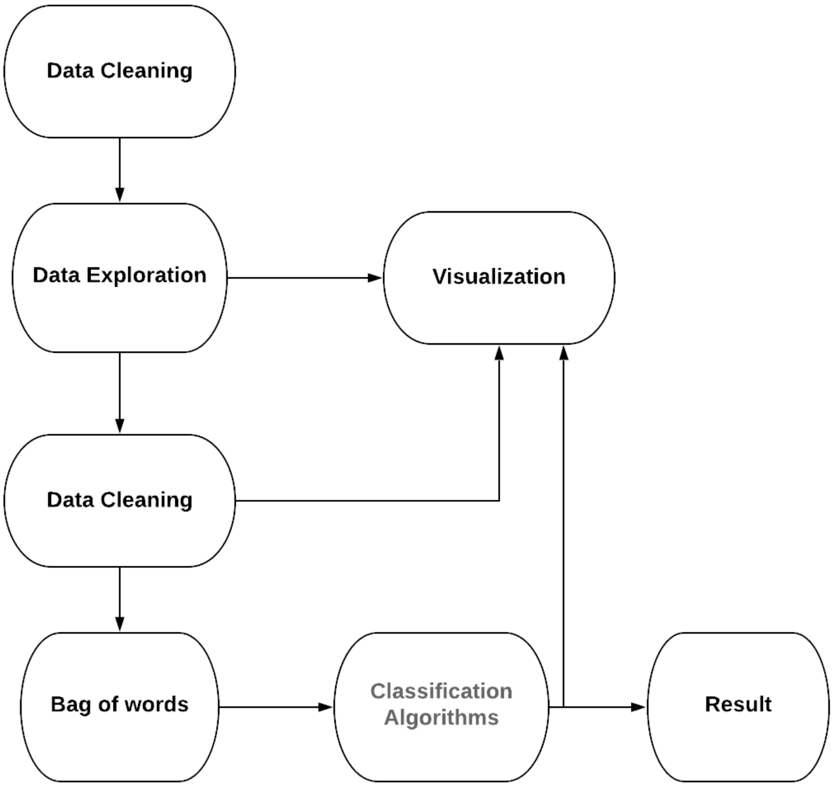


Figure 14 Application implementation Process

Figure 12 showing the step will take to clean our own dataset.

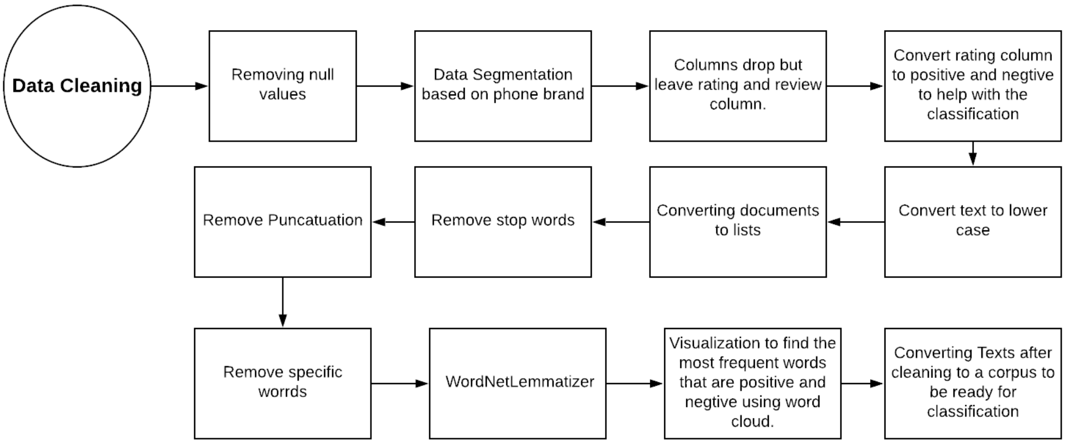

Data Cleaning
Removing null values
Data Segmentation based on phone brand
Columns drop but leave rating and review column.
Convert rating column to positive and negtive to help with the classification
Convert text to lower case
Converting documents to lists
Remove stop words
Remove Puncatuation
Remove specific worrds
WordNetLemmatizer
Visualization to find the most frequent words that are positive and negtive using word cloud.
Converting Texts after cleaning to a corpus to be ready for classification

Figure 15 Data Cleaning Process

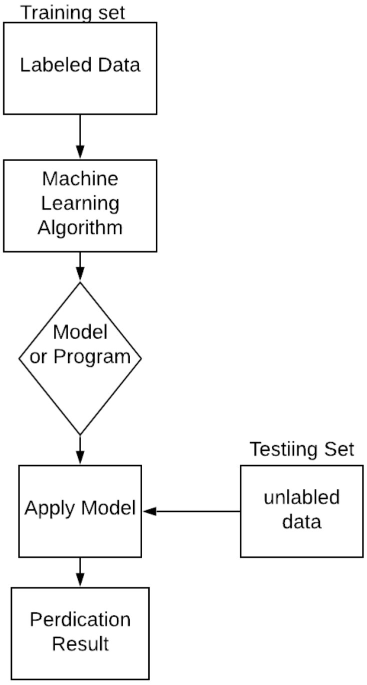


Figure 16 Sentiment Analysis Machine learning Approach for our project

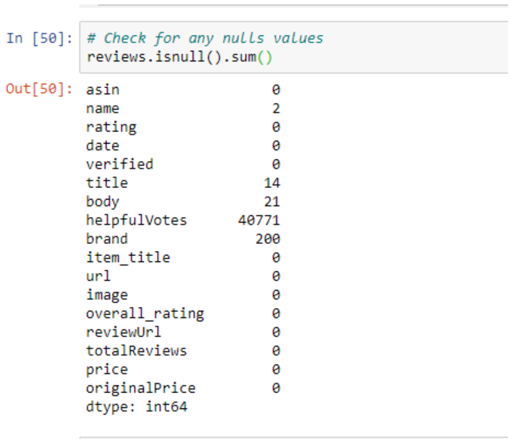
5.2 Implementation

5.2.1 Data Cleaning

We have segmented the data into four data frames based on the brand. This section will include one example from one notebook. Most of the cleaning for the brands is the same, but, of course, receives different results in terms of classification and word cloud.

5.2.1.1 Removing null values

The first step to identify the null values in data ,after merging items and reviews based on the main column. Columns such as body and brand are important for our project. Figure 13 shows null values in both columns that need to be removed.

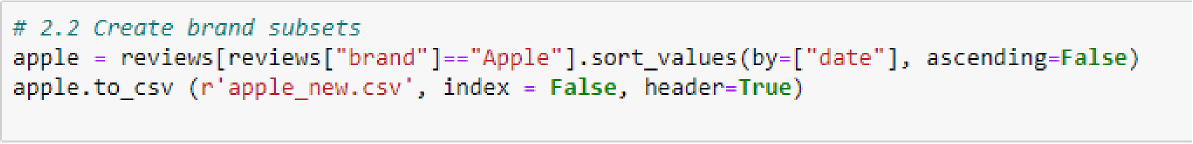


```
In [50]:  # Check for any nulls values
          reviews.isnull().sum()

Out[50]:  asin                0
          name                2
          rating              0
          date                0
          verified            0
          title              14
          body               21
          helpfulVotes    40771
          brand             200
          item_title          0
          url                 0
          image               0
          overall_rating      0
          reviewUrl           0
          totalReviews        0
          price               0
          originalPrice       0
          dtype: int64
```

Figure 17 Removing null values

5.2.1.2 Data Segmentations

The data segmented by the brand column to understand reviews made for different brands to analyze it and get an understanding of which brand most favored on the Amazon website.



```
# 2.2 Create brand subsets
apple = reviews[reviews["brand"]=="Apple"].sort_values(by=["date"], ascending=False)
apple.to_csv (r'apple_new.csv', index = False, header=True)
```

Figure 18 Data Segments Code

5.2.1.3 Dropping unnecessary columns and convert rating result into positive and negative aspect.

The most important columns for our sentiment analysis are the rating and body which have contain customer reviews. Rating is from 1 to 5 in the data file which will be used to label the data so that we can run supervised machine learning algorithm. Based on other literature reviews considered to drop rating 3 which considered as natural then convert 1 and 2 to negative 0 and 4 5 as 1 positive sentiment.

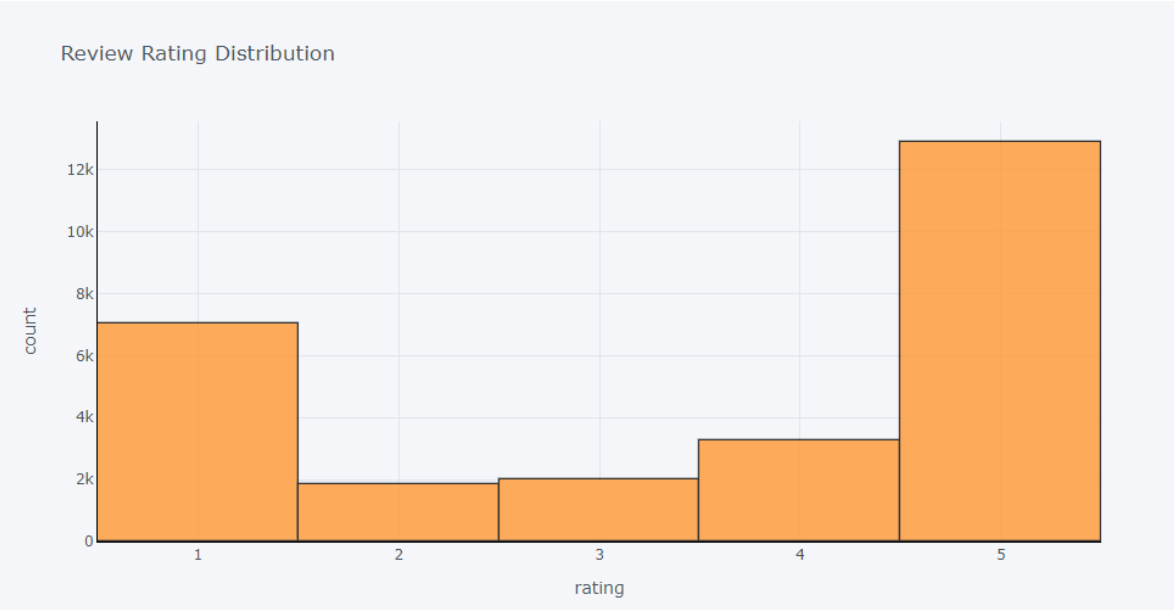
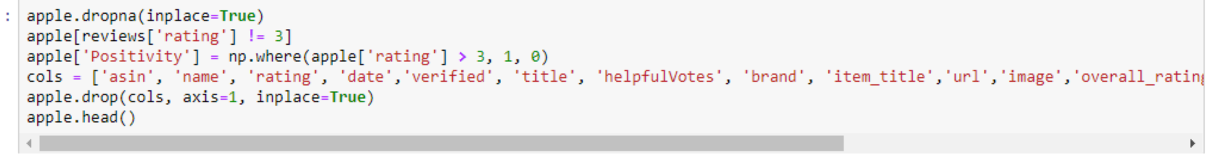
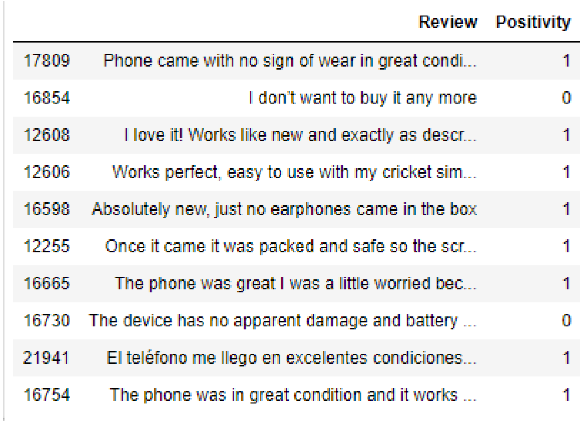


Figure 19 Review Rating Distribution showing A lot of reviews have 5 rating in terms of all the brand



```
apple.dropna(inplace=True)
apple[reviews['rating'] != 3]
apple['Positivity'] = np.where(apple['rating'] > 3, 1, 0)
cols = ['asin', 'name', 'rating', 'date','verified', 'title', 'helpfulVotes', 'brand', 'item_title','url','image','overall_ratin
apple.drop(cols, axis=1, inplace=True)
apple.head()
```

Figure 20 Labelling data code



	Review	Positivity
17809	Phone came with no sign of wear in great condi...	1
16854	I don't want to buy it any more	0
12608	I love it! Works like new and exactly as descr...	1
12606	Works perfect, easy to use with my cricket sim...	1
16598	Absolutely new, just no earphones came in the box	1
12255	Once it came it was packed and safe so the scr...	1
16665	The phone was great I was a little worried bec...	1
16730	The device has no apparent damage and battery ...	0
21941	El teléfono me llego en excelentes condiciones...	1
16754	The phone was in great condition and it works ...	1

Figure 21 Labelled dataset created

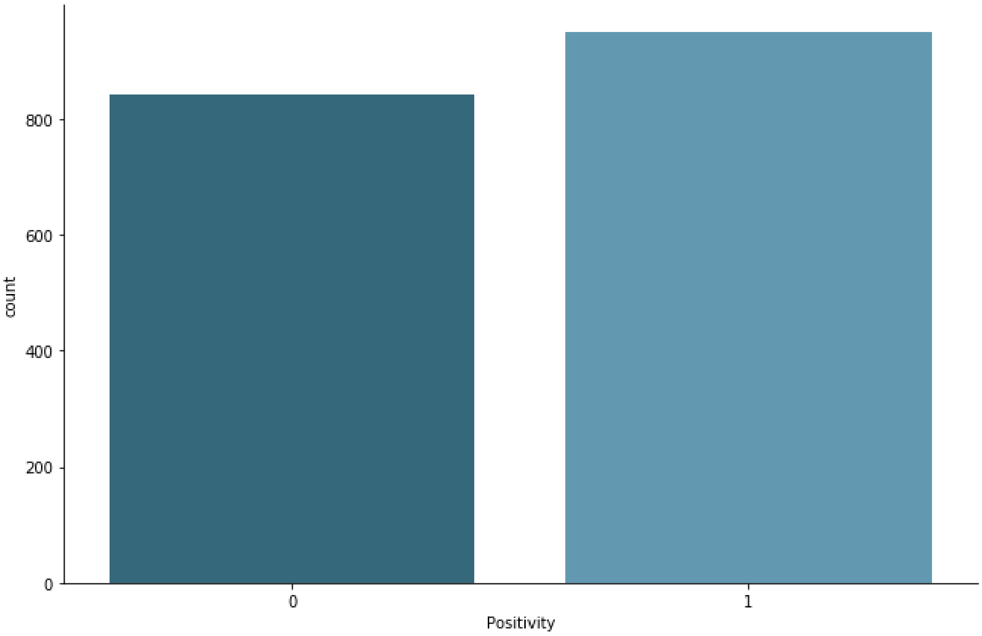
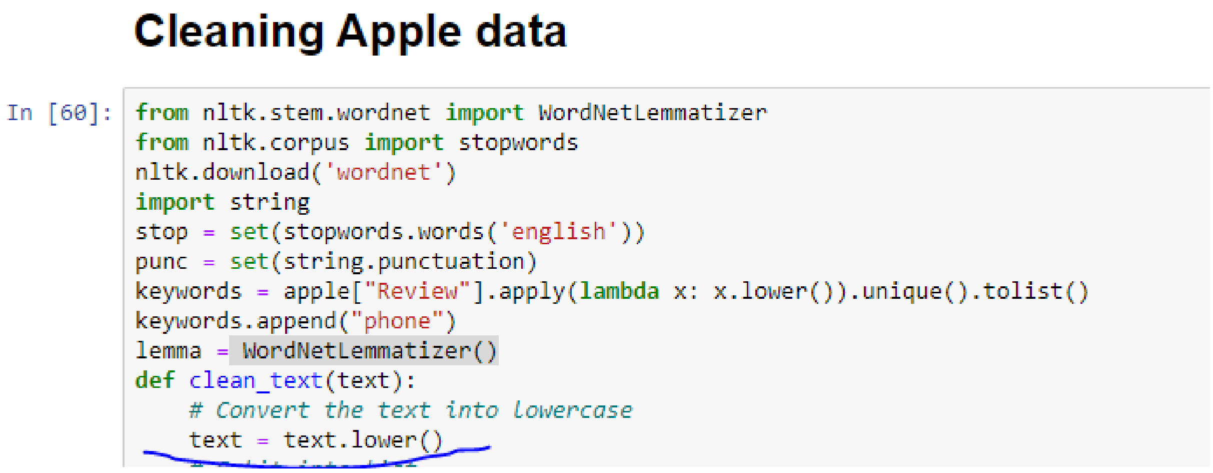


Figure 22 Apple dataset showing to have more slight positive reviews based on the rating

5.2.1.4 Convert text to lower case

Converting all the reviews into the lower case, it’s an important aspect as without might have similar words, one lower case and upper case in the vector created from a bag of words.



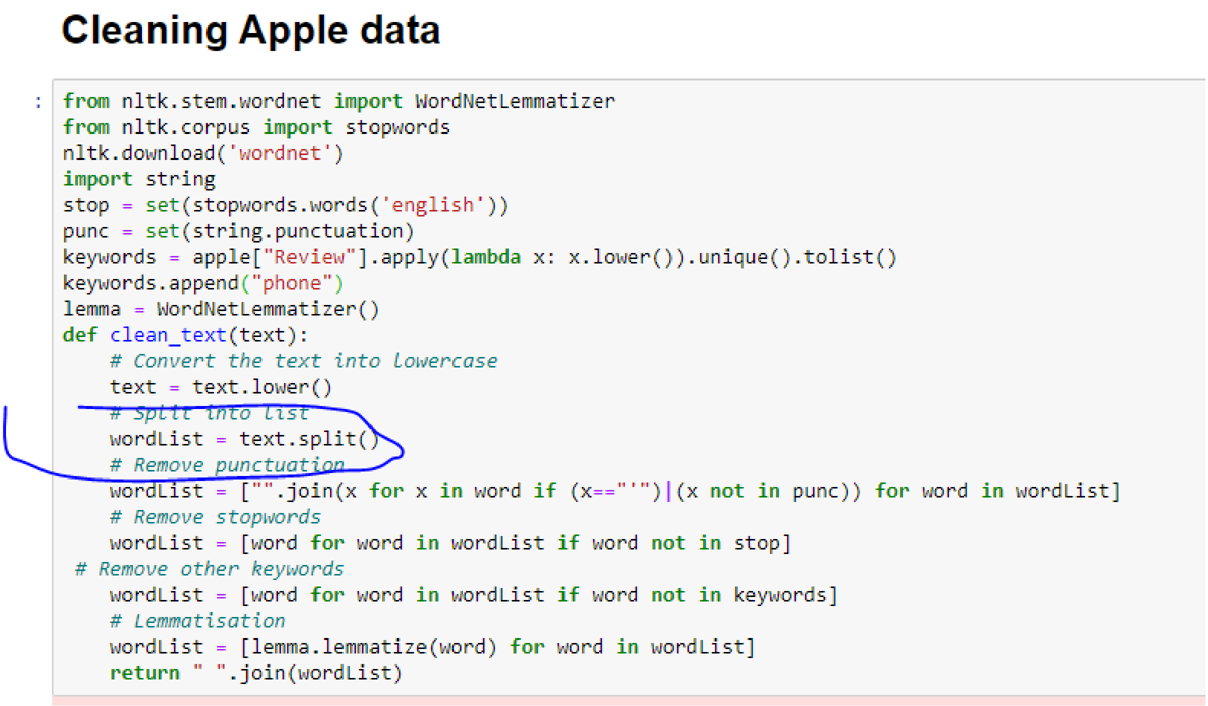
Cleaning Apple data

In [60]:
```
from nltk.stem.wordnet import WordNetLemmatizer
from nltk.corpus import stopwords
nltk.download('wordnet')
import string
stop = set(stopwords.words('english'))
punc = set(string.punctuation)
keywords = apple["Review"].apply(lambda x: x.lower()).unique().tolist()
keywords.append("phone")
lemma = WordNetLemmatizer()
def clean_text(text):
    # Convert the text into lowercase
    text = text.lower()
```

Figure 23 Converting data to lower case

5.2.1.5 *Converting documents into lists*

Converting documents into lists helps to identify stop words, count frequent words and remove duplicates, and so on.



Cleaning Apple data

```
from nltk.stem.wordnet import WordNetLemmatizer
from nltk.corpus import stopwords
nltk.download('wordnet')
import string
stop = set(stopwords.words('english'))
punc = set(string.punctuation)
keywords = apple["Review"].apply(lambda x: x.lower()).unique().tolist()
keywords.append("phone")
lemma = WordNetLemmatizer()
def clean_text(text):
    # Convert the text into lowercase
    text = text.lower()
    # Split into list
    wordList = text.split()
    # Remove punctuation
    wordList = ["".join(x for x in word if (x=="'")|(x not in punc)) for word in wordList]
    # Remove stopwords
    wordList = [word for word in wordList if word not in stop]
 # Remove other keywords
    wordList = [word for word in wordList if word not in keywords]
    # Lemmatisation
    wordList = [lemma.lemmatize(word) for word in wordList]
    return " ".join(wordList)
```

Figure 24 Converting documents into a list

5.2.1.6 Stop words

Removing stop words was done using nltk stop words such as “is , the,a” etc from the reviews because they don’t carry important information. Figure 17

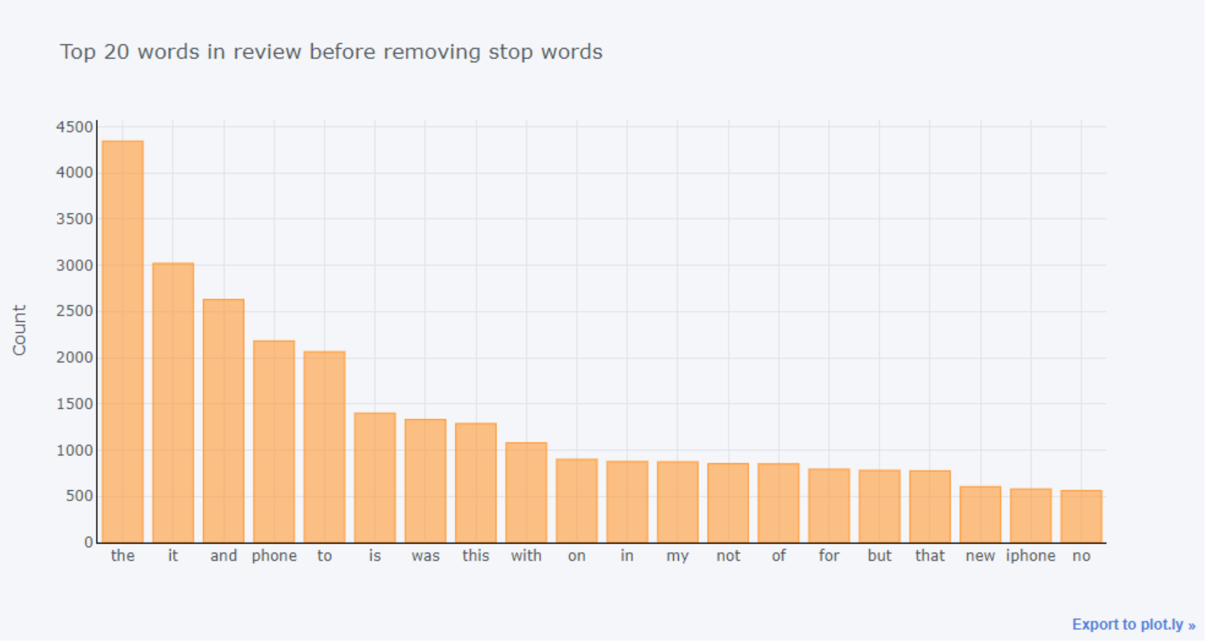
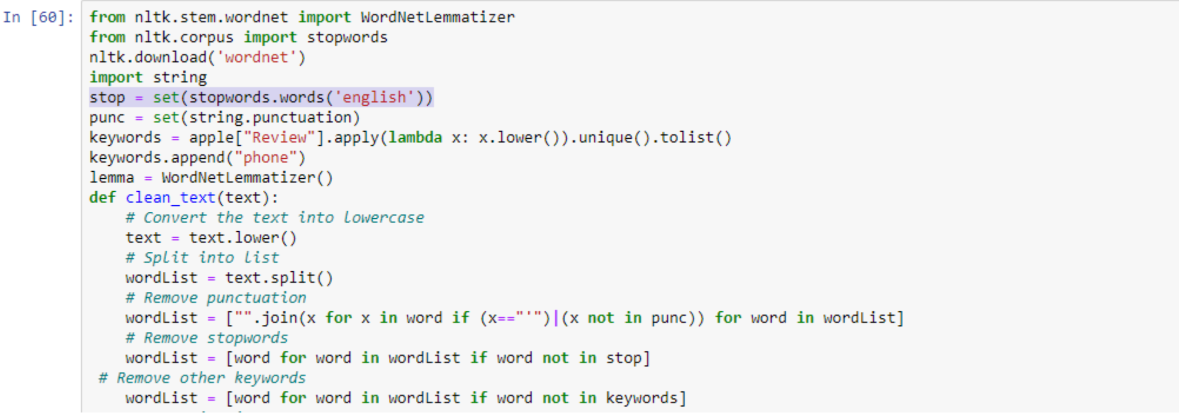


Figure 26 Stop Words identified in the reviews



```
In [60]: from nltk.stem.wordnet import WordNetLemmatizer
         from nltk.corpus import stopwords
         nltk.download('wordnet')
         import string
         stop = set(stopwords.words('english'))
         punc = set(string.punctuation)
         keywords = apple["Review"].apply(lambda x: x.lower()).unique().tolist()
         keywords.append("phone")
         lemma = WordNetLemmatizer()
         def clean_text(text):
             # Convert the text into lowercase
             text = text.lower()
             # Split into list
             wordList = text.split()
             # Remove punctuation
             wordList = ["".join(x for x in word if (x=="'")|(x not in punc)) for word in wordList]
             # Remove stopwords
             wordList = [word for word in wordList if word not in stop]
          # Remove other keywords
             wordList = [word for word in wordList if word not in keywords]
```

Figure 25 Stop words code

After removing the stop word, we run our function again to check the result. Figure 19 shows most of the stop words have been removed.

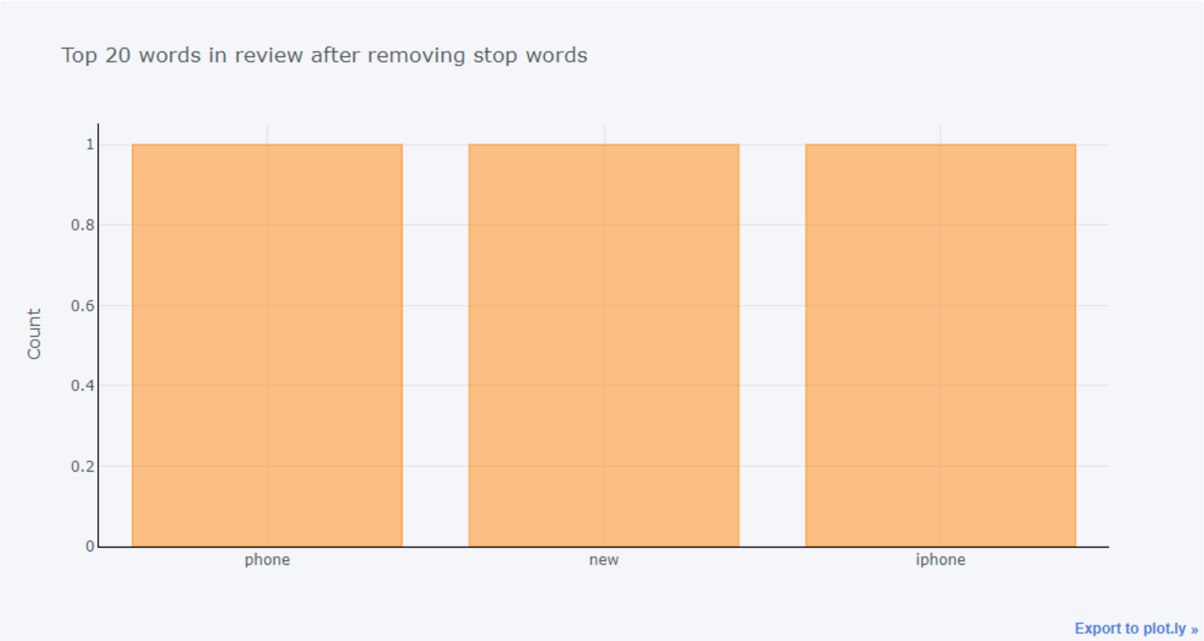
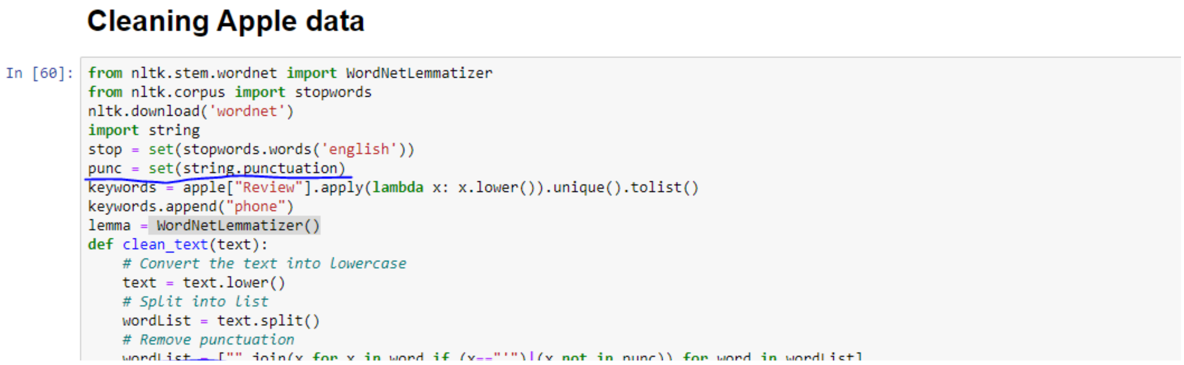


Figure 27 Result after removing stop words

5.2.1.7 Removing punctuation

Removing punctuation helps to get the root of a text, so its a very important aspect to be removed.



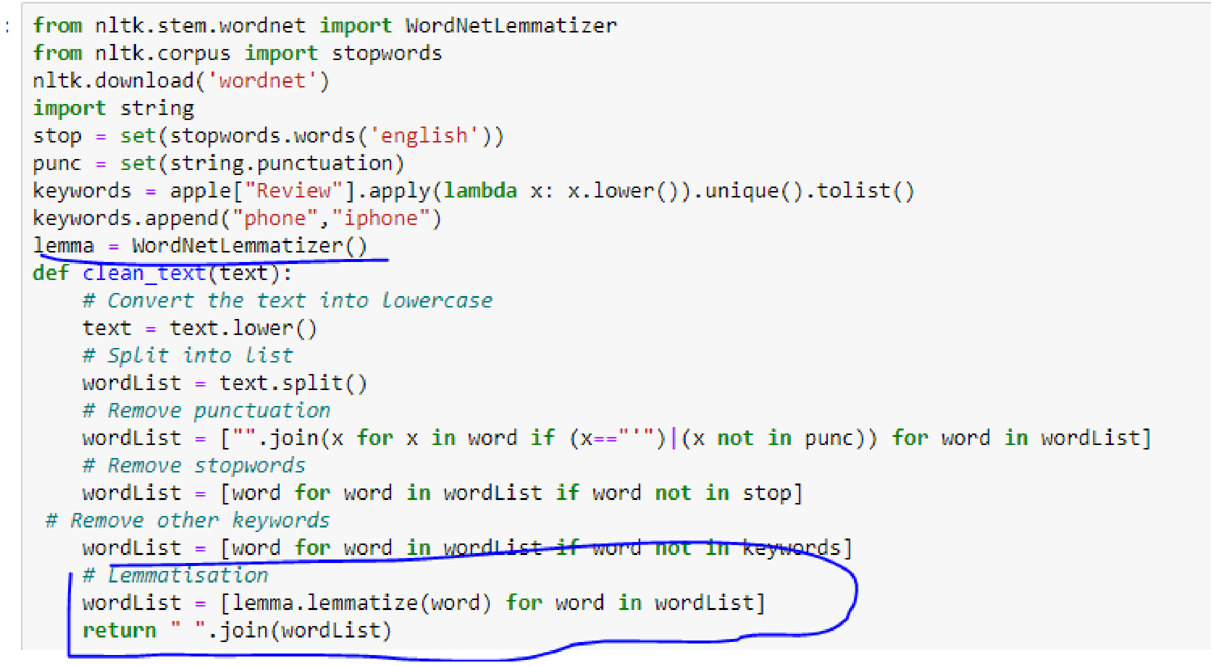
Cleaning Apple data

```
In [60]: from nltk.stem.wordnet import WordNetLemmatizer
         from nltk.corpus import stopwords
         nltk.download('wordnet')
         import string
         stop = set(stopwords.words('english'))
         punc = set(string.punctuation)
         keywords = apple["Review"].apply(lambda x: x.lower()).unique().tolist()
         keywords.append("phone")
         lemma = WordNetLemmatizer()
         def clean_text(text):
             # Convert the text into lowercase
             text = text.lower()
             # Split into list
             wordList = text.split()
             # Remove punctuation
             wordList = ["".join(x for x in word if (x=="'")|(x not in punc)) for word in wordList]
```

Figure 28 Remove Punctuation

5.2.1.8 Lemmatize

Lemmatize function is like stemming, but it brings context to the words which link words with similar meaning to one word. The function helps to get the root of the texts in each review. Lemmatize does things properly with use if a vocabulary and morphological analysis of words and aims to delete inflectional endings only and return the base form of a word. Lemmatize is better than stemming because stemming referred to as a crude heuristic process that chops of the end of the words without the knowledge of the context. Chopping the words without the knowledge makes not to discriminate between words that have different meanings depending on the part of the speech. It’s still preferred that stemming is quite easy to implement and run faster, and the reduced accuracy may not matter for applications.



```
from nltk.stem.wordnet import WordNetLemmatizer
from nltk.corpus import stopwords
nltk.download('wordnet')
import string
stop = set(stopwords.words('english'))
punc = set(string.punctuation)
keywords = apple["Review"].apply(lambda x: x.lower()).unique().tolist()
keywords.append("phone","iphone")
lemma = WordNetLemmatizer()
def clean_text(text):
    # Convert the text into lowercase
    text = text.lower()
    # Split into list
    wordList = text.split()
    # Remove punctuation
    wordList = ["".join(x for x in word if (x=="'")|(x not in punc)) for word in wordList]
    # Remove stopwords
    wordList = [word for word in wordList if word not in stop]
 # Remove other keywords
    wordList = [word for word in wordList if word not in keywords]
    # Lemmatisation
    wordList = [lemma.lemmatize(word) for word in wordList]
    return " ".join(wordList)
```

Figure 29 Lemmatize code

5.2.1.9 Visualization to find out the most frequent words that are positive and negative using a word cloud.

Figure 26 showing the latest 1000 positive based on the frequency. Customer showing to like the iPhone battery, camera, charger, and phone screen in terms of the feature. People showed emotion as well by writing the word happy, life, perfect, and like but also mentioned about buying unlocked phones.

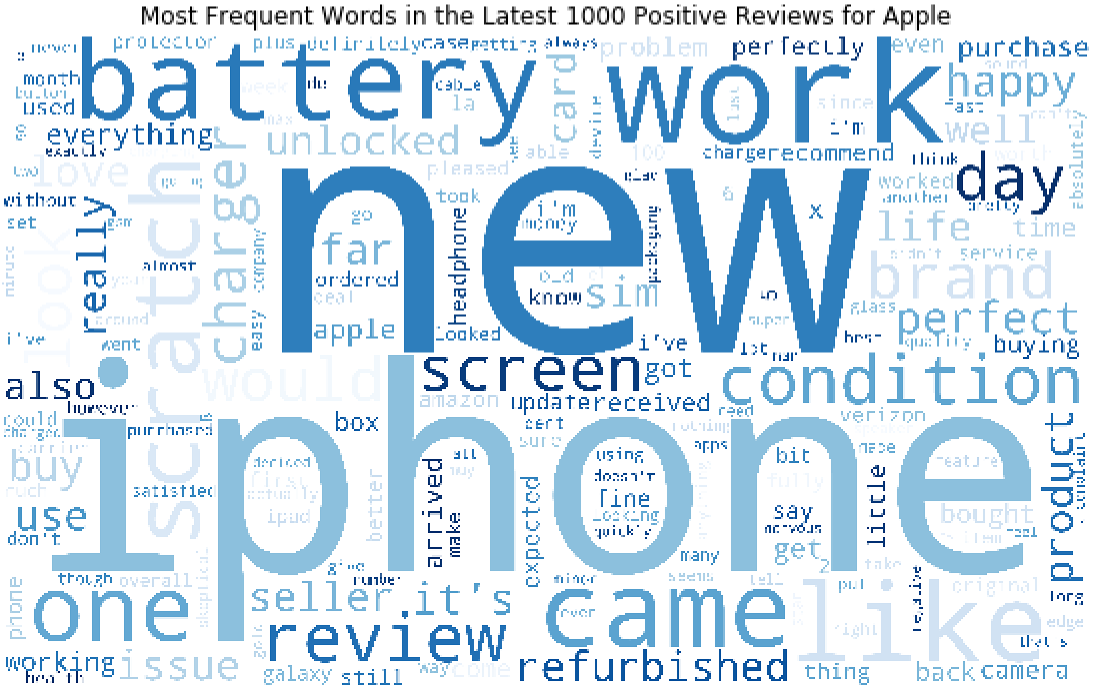


Figure 30 Most frequent words that classed as positive based on frequency

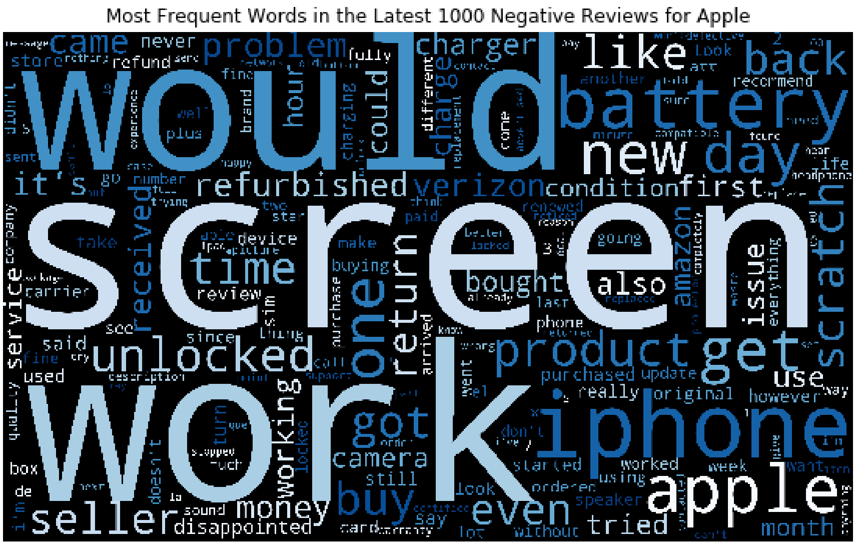


Figure 31 Most Frequent negative words for Apple Reviews

Figure 27 showing the most frequent negative words written about apple phones. Some disagree with other reviewers that the battery of the Apple phones is not great and similar to the screen and battery. Emotion also showed such disappointment, tried, etc.

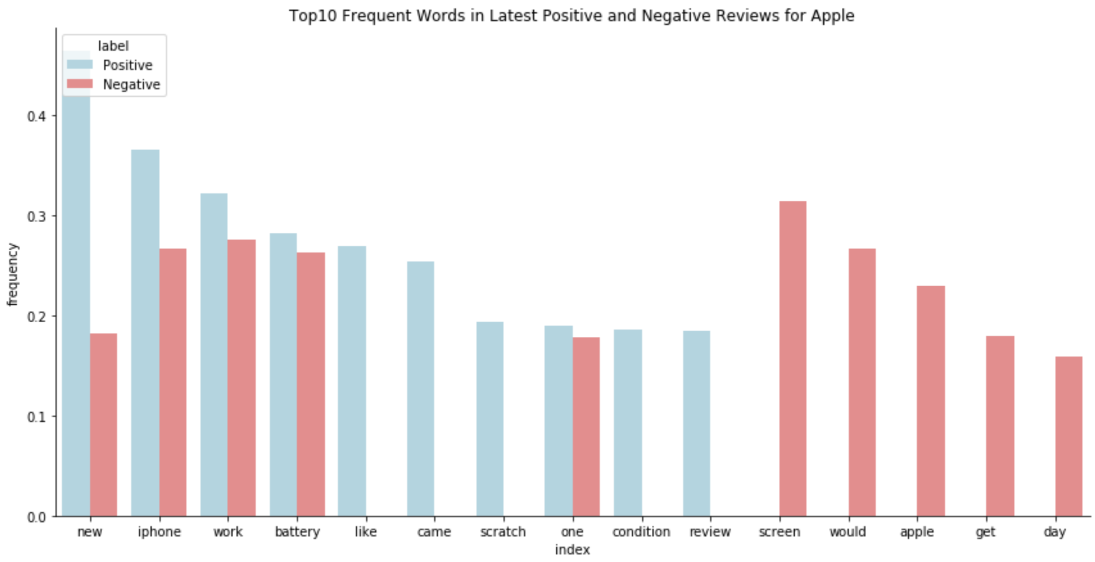
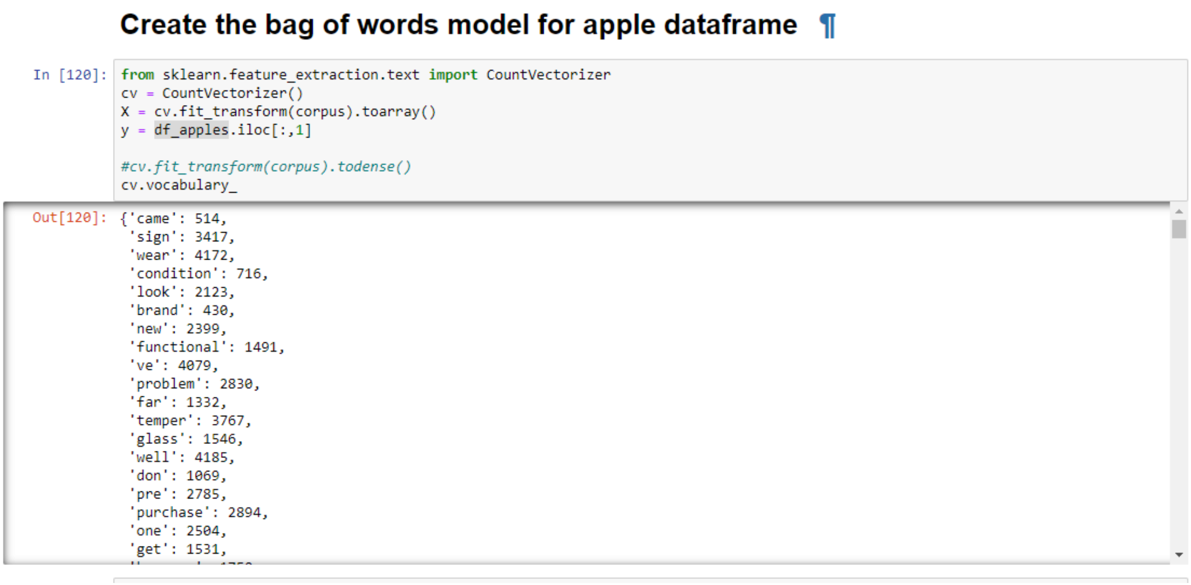


Figure 32 Visualizing both negative and positive words

5.2.2 Bag of Words



Create the bag of words model for apple dataframe ¶

```
In [120]: from sklearn.feature_extraction.text import CountVectorizer
          cv = CountVectorizer()
          X = cv.fit_transform(corpus).toarray()
          y = df_apples.iloc[:,1]

          #cv.fit_transform(corpus).todense()
          cv.vocabulary_
```

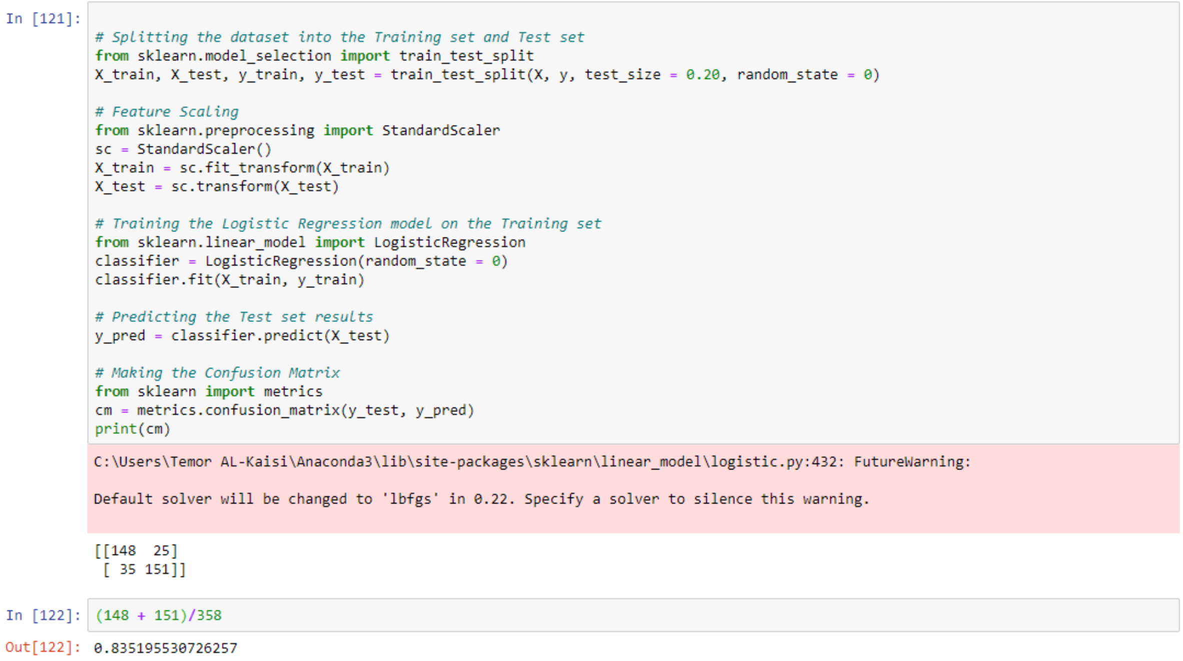
```
Out[120]: {'came': 514,
           'sign': 3417,
           'wear': 4172,
           'condition': 716,
           'look': 2123,
           'brand': 430,
           'new': 2399,
           'functional': 1491,
           've': 4079,
           'problem': 2830,
           'far': 1332,
           'temper': 3767,
           'glass': 1546,
           'well': 4185,
           'don': 1069,
           'pre': 2785,
           'purchase': 2894,
           'one': 2504,
           'get': 1531,
```

Figure 33 Bag of words code and result

5.2.3 Classification Algorithm

5.2.3.1 Logistic Regression

The approach used to logistic regression algorithm is similar steps used with other algorithms that will be identified in the visualization section. Firstly, split the data set into training and test set by 80% training and 20% for test data set. I also used a feature scaling method to standardize the independent features presents in the data in a fixed range. The training dataset on the logistic regression model then predicts the test set result and out of that make the confusion matrix, which is a table that describes the performance of a classifier onset of test data.



```
In [121]:
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# Training the Logistic Regression model on the Training set
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn import metrics
cm = metrics.confusion_matrix(y_test, y_pred)
print(cm)
```

```
C:\Users\Temor AL-Kaisi\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning:

Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.

[[148  25]
 [ 35 151]]
```

```
In [122]: (148 + 151)/358
Out[122]: 0.835195530726257
```

Figure 34 Commented code for logistic result

The result of the confusion metrics implemented as heat map in figure 33.

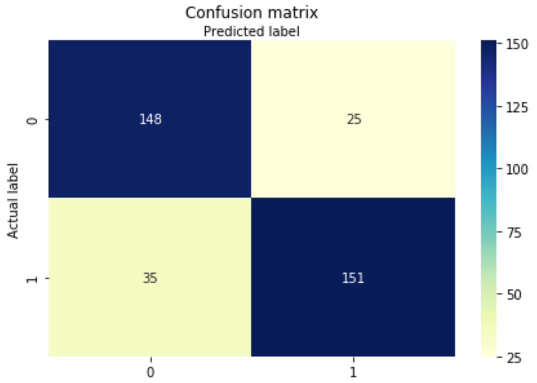
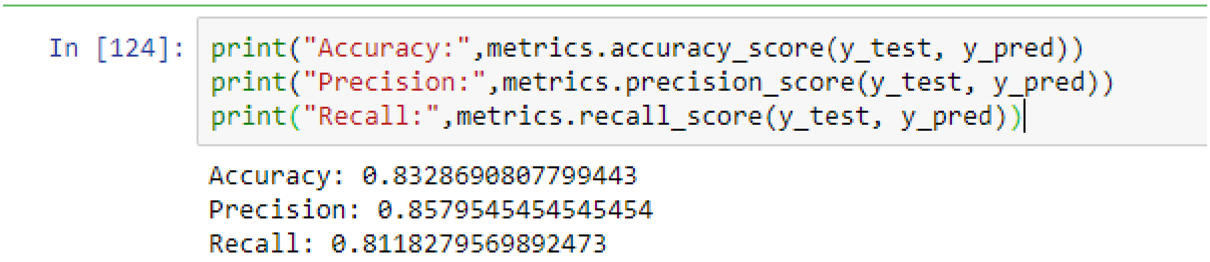


Figure 35 Classification algorithm result of the confusion metric as a heat map



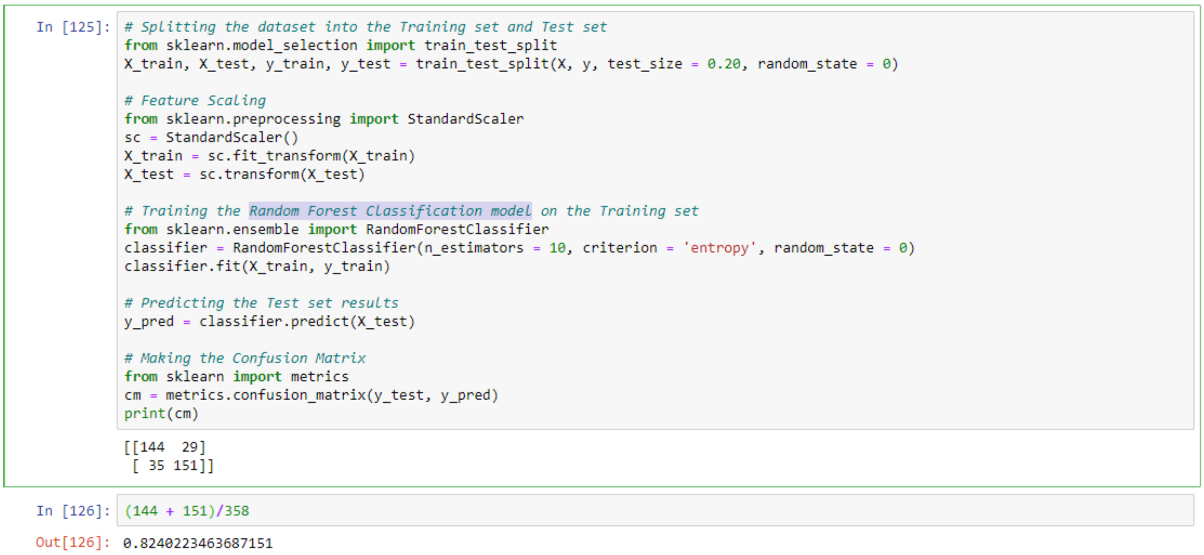
```
In [124]: print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
          print("Precision:",metrics.precision_score(y_test, y_pred))
          print("Recall:",metrics.recall_score(y_test, y_pred))

          Accuracy: 0.8328690807799443
          Precision: 0.8579545454545454
          Recall: 0.8118279569892473
```

Figure 36 Accuracy, Precision and Recall result of logistic regression algorithm

The result of the logistic regression model showing to be good result over 80 for accuracy, precision, and recall.

5.2.3.2 Random Forest Classification model

Random forest scored less accuracy result by 1% for the apple dataset than the logistic regression model, which 82% where LR scored 83% for accuracy.



```
In [125]: # Splitting the dataset into the Training set and Test set
          from sklearn.model_selection import train_test_split
          X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)

          # Feature Scaling
          from sklearn.preprocessing import StandardScaler
          sc = StandardScaler()
          X_train = sc.fit_transform(X_train)
          X_test = sc.transform(X_test)

          # Training the Random Forest Classification model on the Training set
          from sklearn.ensemble import RandomForestClassifier
          classifier = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 0)
          classifier.fit(X_train, y_train)

          # Predicting the Test set results
          y_pred = classifier.predict(X_test)

          # Making the Confusion Matrix
          from sklearn import metrics
          cm = metrics.confusion_matrix(y_test, y_pred)
          print(cm)

          [[144  29]
           [ 35 151]]

In [126]: (144 + 151)/358

Out[126]: 0.8240223463687151
```

Figure 37 Commented Random Forest Algorithm

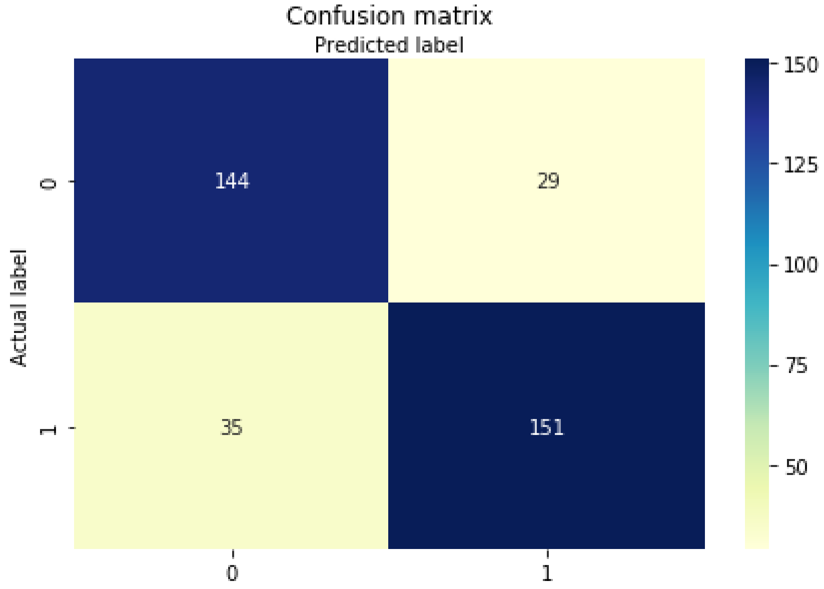
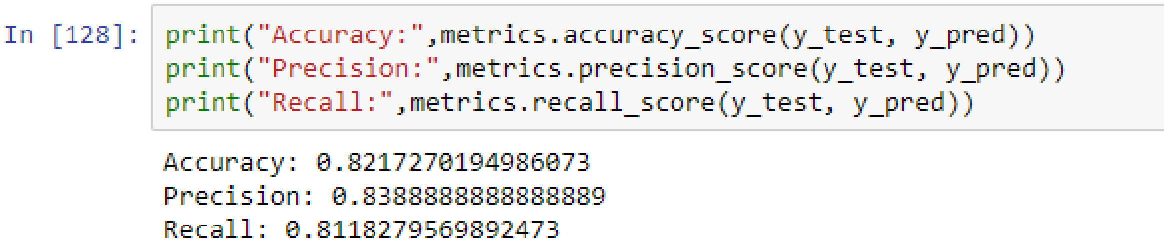


Figure 38 Random Forest Algorithm Confusion Matrix result as heatmap



```
In [128]: print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
          print("Precision:",metrics.precision_score(y_test, y_pred))
          print("Recall:",metrics.recall_score(y_test, y_pred))
```

```
Accuracy: 0.8217270194986073
Precision: 0.8388888888888889
Recall: 0.8118279569892473
```

Figure 39 Results in Percentages

6. Visualisations

6.1 Data Exploring

6.1.1 Tableau Dashboard

Online tableau dashboard online link:

https://public.tableau.com/profile/mohamed.alkaisi#!/vizhome/tablueawork/AmazonCellphoneReviews

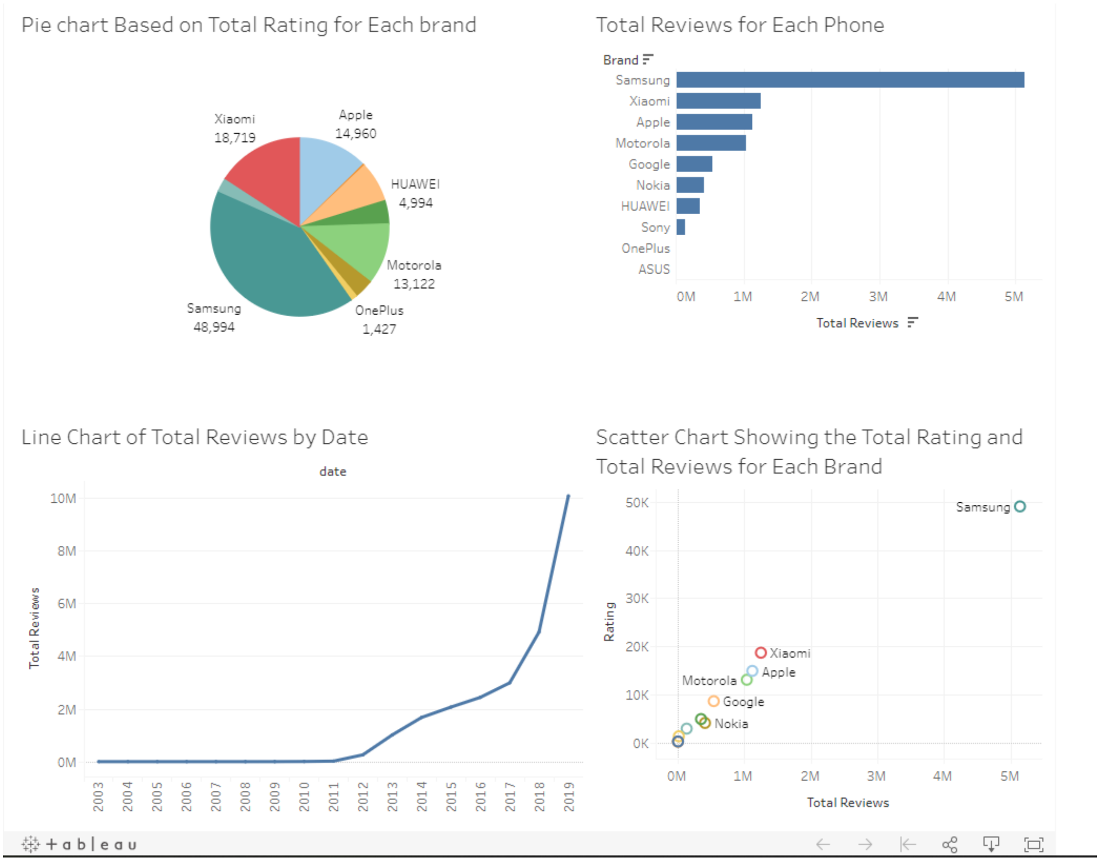


Figure 40 Tableau dashboard exploring the dataset

The bar chart in the dashboard showing Samsung to have the highest total of reviews. The bar chart showing Samsung to have the highest total rating. The scatter chart showing that Samsung to have the highest rating and total reviews. The line chart showing the reviews created during the year and that reviews created in our dataset from 2011 until 2019 but mostly created in 2019.

6.2 Samsung

As we have already explained that the data cleaning for all the brands is the same. Already explained cleaning data for Apple, then in this section, we just display the word clouds and line plot with a classification algorithm.   Figure 38 showing Samsung based on the rating has mostly positive reviews.

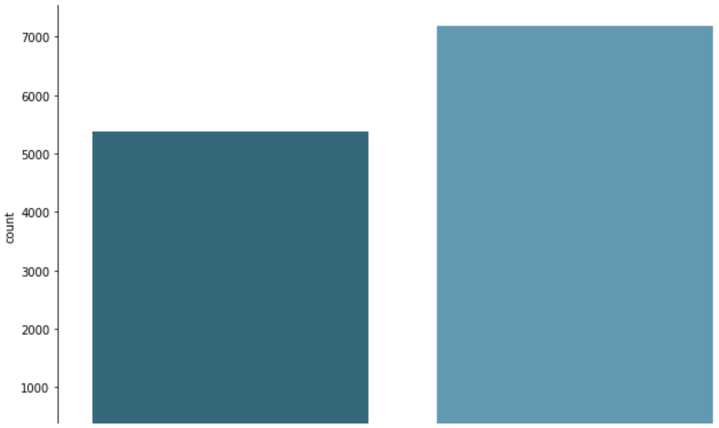


Figure 41 Comparing positive reviews vs negative reviews based on rating

Figure 39 showing positive words mentioned by customers about Samsung mobile features such as screen, battery, pictures, and the price. Emotion also showed, such as love, life, looking, happy, etc.

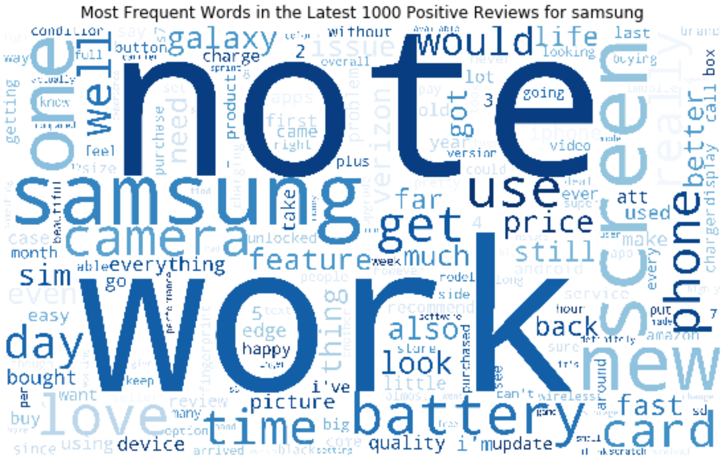


Figure 42 Most Frequent positive words for Samsung

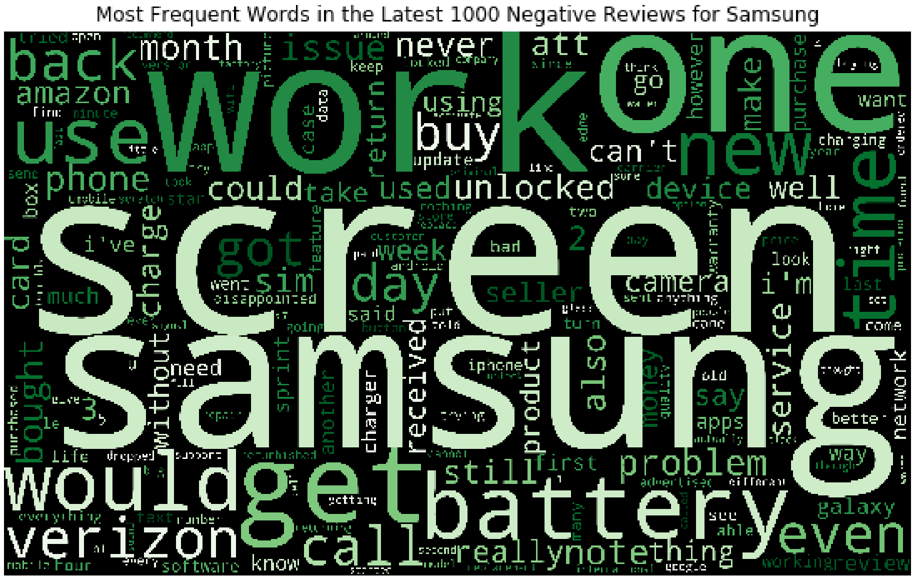


Figure 43 Most Frequent negative words for Samsung

Figure 39 showing the most frequent negative words for Samsung. The word cloud is showing negative words about the features and their own emotion. Customers like apple also disagreed with other people who mentioned positive features about the screen, battery, charge. Their emotion also extracted, such as disappointed etc.

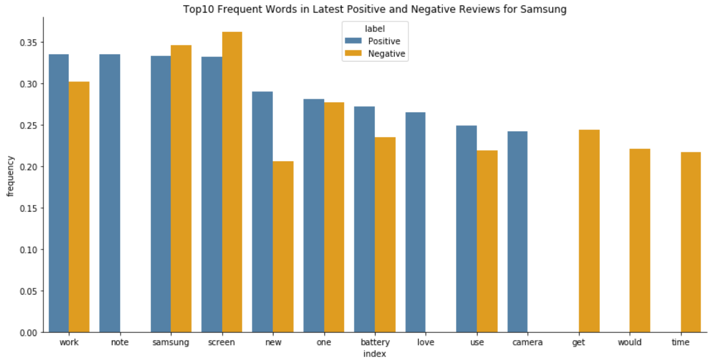


Figure 44 Top 10 Frequent words in Latest positive and negative reviews for Samsung

Based on the frequency figure, 40 showing the top positive and negative words for Samsung phone. Some words could be cleaned from the dataset based on figure 40, such as “one,” use,” get,” and ”would.” This would improve our classification accuracy.

Random Forest Algorithm

Random forest Algorithm gave the highest accuracy results for the Samsung dataset of 81% than logistic regression by 1% higher. The data split was 20% of data for test and 80% for training.

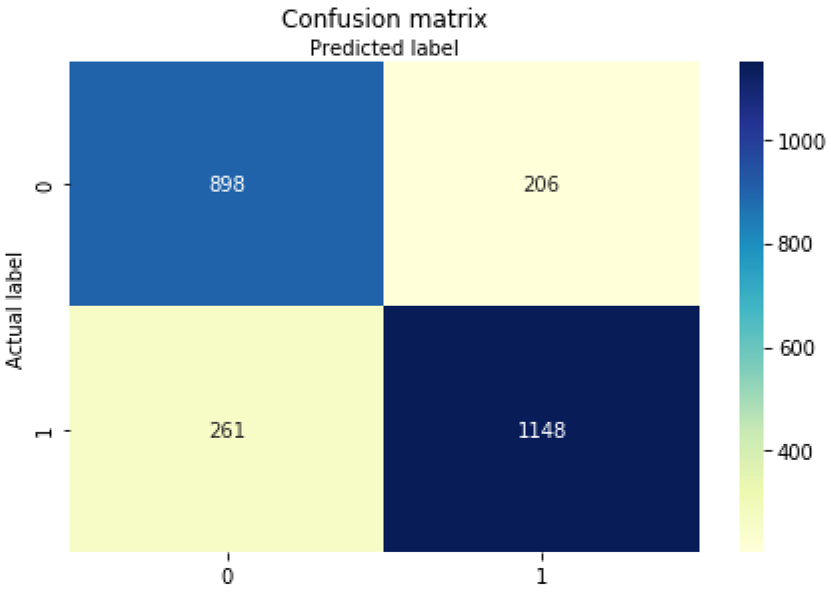
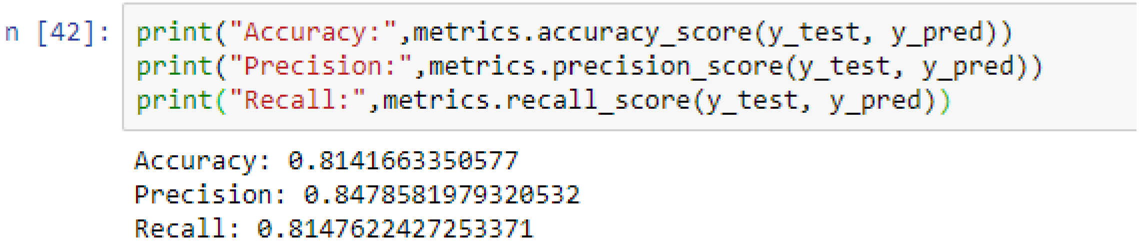


Figure 45 Confusion Meteric results fior Random Forest algorithm



```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
print("Precision:",metrics.precision_score(y_test, y_pred))
print("Recall:",metrics.recall_score(y_test, y_pred))
```

```
Accuracy: 0.8141663350577
Precision: 0.8478581979320532
Recall: 0.8147622427253371
```

Figure 46 Random Forest Algorithm result

Logistic Regression model

Logistic Regression Algorithm gave one percent lower than random forest but still better than Naive Bayes or SVM as both results were less than 80%. The data split was 20% of data for test and 80% for training.

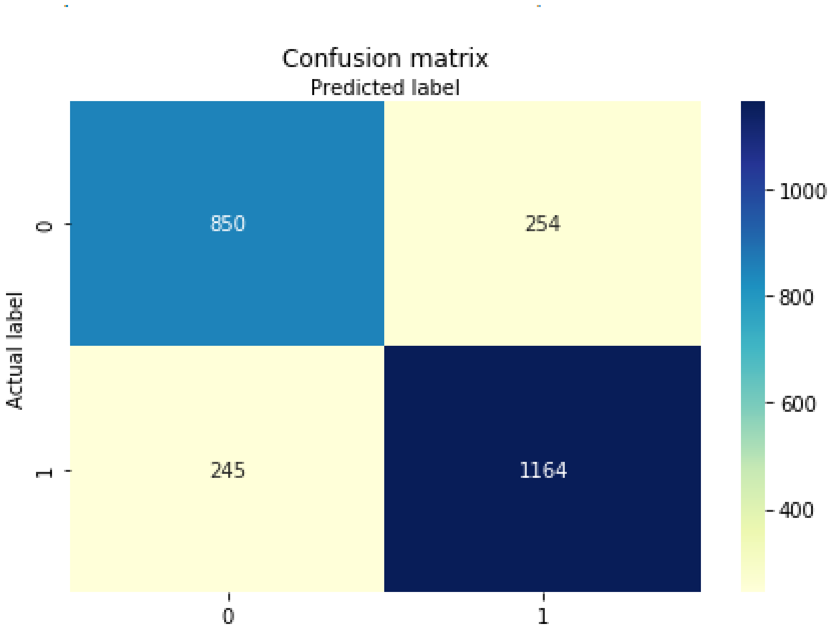
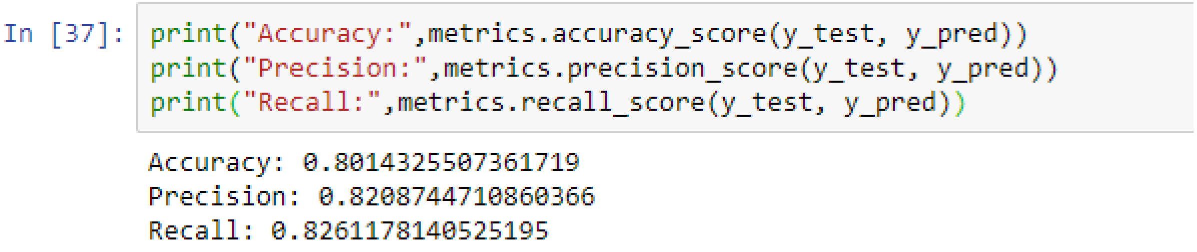


Figure 47 Confusion Martix showing results for logistic regression model



```
In [37]: print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
         print("Precision:",metrics.precision_score(y_test, y_pred))
         print("Recall:",metrics.recall_score(y_test, y_pred))
```

```
Accuracy: 0.8014325507361719
Precision: 0.8208744710860366
Recall: 0.8261178140525195
```

Figure 48 Logistic regression result based on metric showing accuracy, precision and recall

6.3 Motorola

Motorola brand showing to be the second brand to have the highest rating after Samsung and highest total reviews after

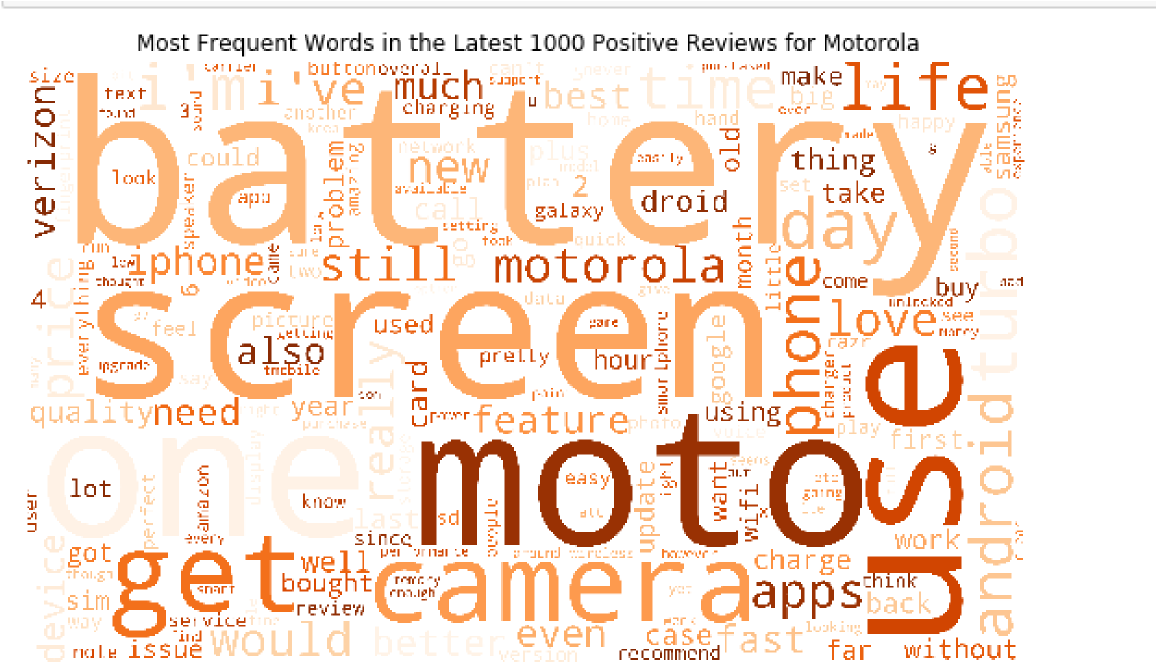
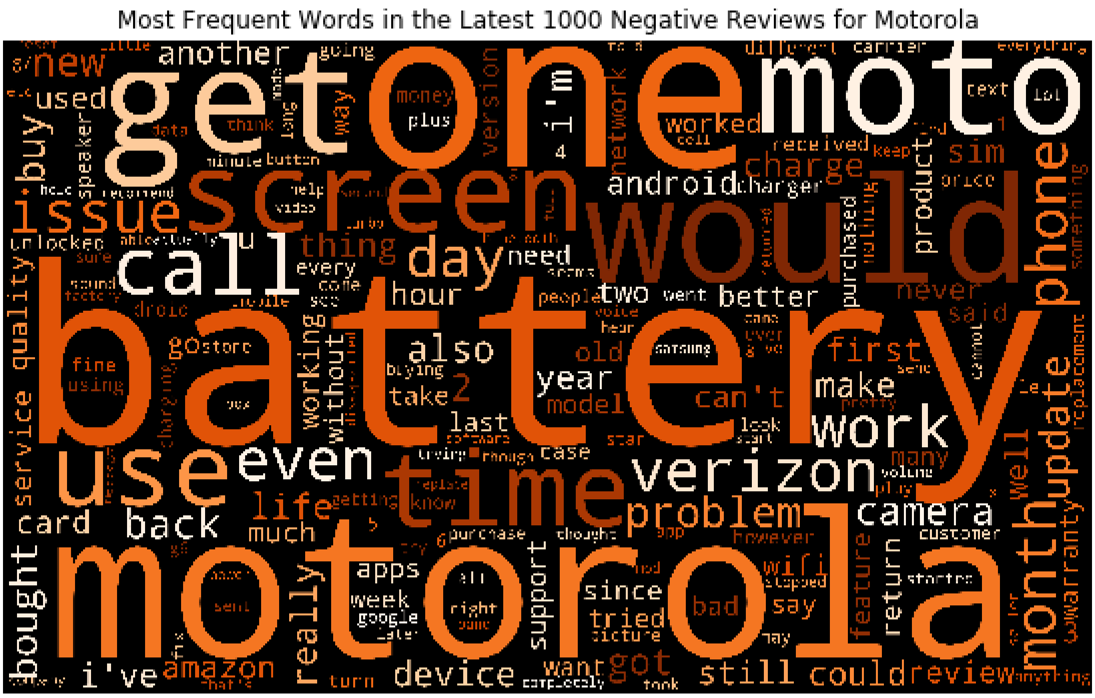

Most Frequent Words in the Latest 1000 Positive Reviews for Motorola

Figure 49 Most Frequent positive words for Motorola phone

Figure 42 showing the most frequent positive words for Motorola brand. Again the word cloud is showing positive words about the phone feature and similar to Samsung and Apple, showing that Motorola received positive about their own screen, battery, and a camera, etc.   Emotion words also extracted, such as love and happy etc.



Most Frequent Words in the Latest 1000 Negative Reviews for Motorola

Figure 50 Most frequent negative words about Motorola

Figure 43 showing the most frequent negative words for Motorola. The word cloud is showing negative words about the features and their own emotion. Customers like apple, Samsung also disagreed with other people who mentioned positive features about the screen, battery, charge. Their emotion also extracted, such as disappointed etc.

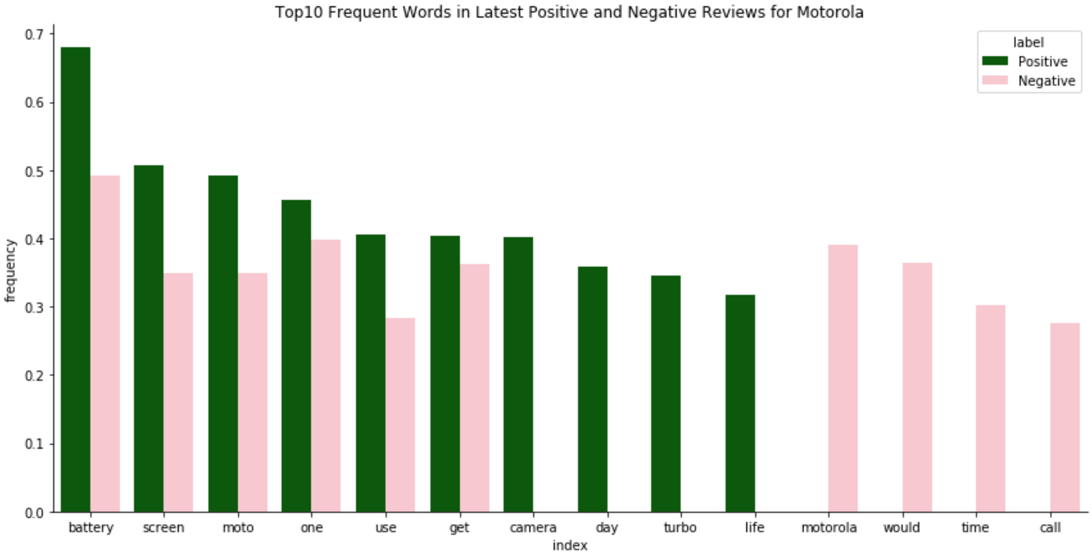


Figure 51 Top 10 frequent words in Latest positive and negative reviews for Motorola

Based on the figure 44 identified that in terms of the top 10 negative and positive there are some words that are meaningful such as get or use or would which should be removed. The most feature received the highest positive score in figure 44 is the battery, and at least negative feature is a call referring to calling.

Kernel SVM model

Kernal SVM algorithm did not show a great result, and it was a 66% accuracy compared to the Random forest algorithm showed a result of 80% accuracy.

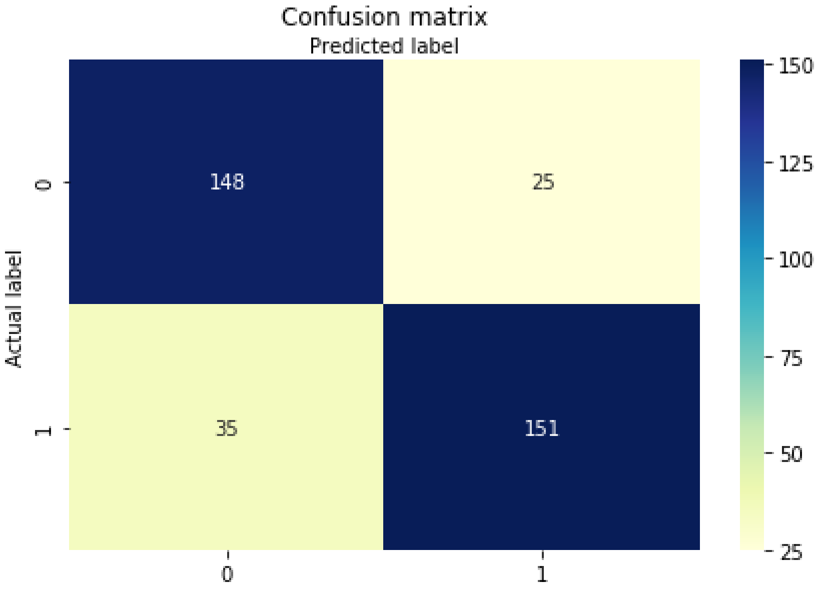
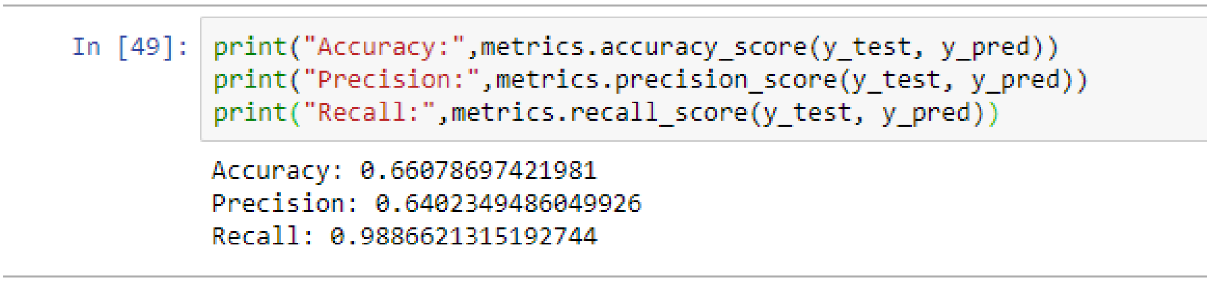


Figure 52 Confusion matrix result for SVM based on Motorola



```
In [49]: print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
         print("Precision:",metrics.precision_score(y_test, y_pred))
         print("Recall:",metrics.recall_score(y_test, y_pred))

         Accuracy: 0.66078697421981
         Precision: 0.6402349486049926
         Recall: 0.9886621315192744
```

Figure 53 result

Random Forest Algorithm

Random forest Algorithm gave the highest accuracy results for the Motorola dataset of 80% than Kernal SVM, Decision Tree, and Naïve Bayes. The data split was 20% of data for test and 80% for training.

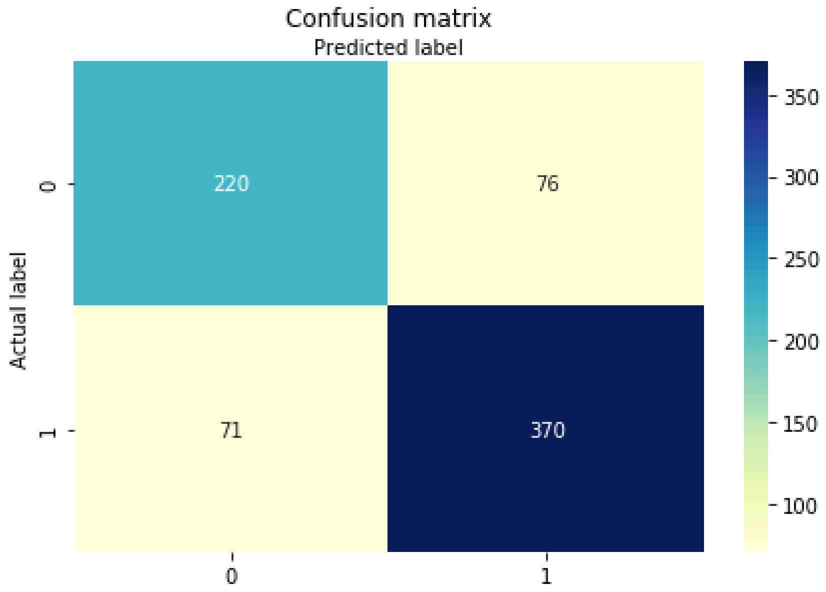
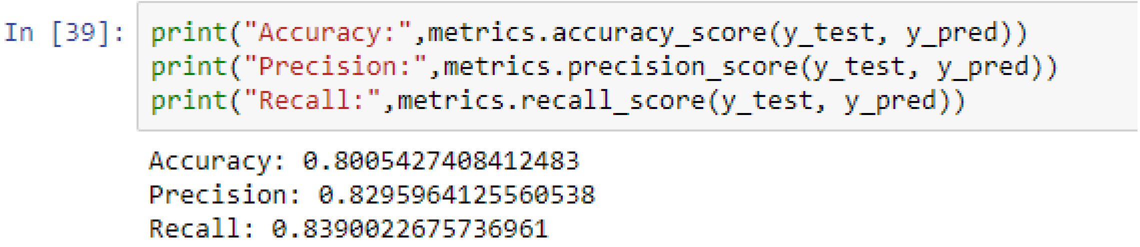


Figure 54 Confusion Mertic result for Random Forest based on Motorola



```
In [39]: print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
         print("Precision:",metrics.precision_score(y_test, y_pred))
         print("Recall:",metrics.recall_score(y_test, y_pred))

         Accuracy: 0.8005427408412483
         Precision: 0.8295964125560538
         Recall: 0.8390022675736961
```

Figure 55 Result In terms of Accuracym precison and recall for Random Forest algorithm based on Motorola dataset

7. Results

The result based on the data exploring at the first stage was based on rating or total reviews that Samsung is number talked about by Amazon customers. It was also estimated by the visualization and conversion of the rating column that Samsung has the most positive result. Motorola brand came in second after Samsung has the most positive reviews.

Based on the word cloud, The story from all the three brands showing in the report such as apple, Samsung and Motorola that people have talked about features of phone positively but the same words repeated in the negative word cloud which explaining that the result based on word cloud are equal in terms the positive word mentioned and negative word mentioned.

The result from running a logistic regression classification against the apple dataset showed higher results than the use of the random forest algorithm. However, running the same algorithm in the Samsung dataset, which is bigger in size than Apple, showed the opposite result in which the random forest algorithm scored higher results than logistic regression. Running the same algorithm on Motorola dataset logistic regression gave less result than the Random forest algorithm, then I have tried Kernal SVM and gave less accurate results than Random forest.

Random forest algorithm showing to display higher accuracy results Motorola and Samsung. Logistic regression scored higher with the apple dataset.

8. Discussion and evaluation.

The chosen industry for this project was the online retail industry from the requirement. The data mining subject used was sentiment analysis using supervised machine learning. The requirement was to build a data warehouse for the chosen data set. The university Oracle database tablespace quota allocation does not allow to store dataset with 27000 rows as small allocation given for each student. The main data column for sentiment analysis was the reviews, and it has characters of more than 4000 characters that exceed varchar2 data type in Oracle. As the data warehouse was a requirement, I had to remove the review column and keep the rest of the data then minimize it into 200 rows from 27000. I had also had to denormalize the data to only three tables for DW, so I do not exceed the limit of table space quotas. DW created can help to identify which brand has the total reviews, which month or year was the best for people to buy phones looking the date where reviews are written and so on.

Data cleaning is the most important stage for sentiment analysis as classification results can be affected by uncleaned data such as data with stop words or duplicate words added because one written as positive and second was written as negative. After cleaning the data using the different features, I still can see the data had words with no meaning, such as phone or create and so on. If I wanted to get a higher accuracy result from classification, then I would look at the cleaning stage back.

However, the system implemented complete the aim and objective of the literature. The first aim to understand what type of brand was most talked about in Amazon, and it identified is Samsung, then Motorola by 2019 following all different stage and classification algorithm showing a similar result. The second aim was to understand how customers have expressed themselves in regard to phone features and their feelings towards the brand, which showed during our visualization using a word cloud.

9. Conclusions and any directions for future work

Unfortunately, due to our university database limitation could not load most of the data into our data warehouse. However, an ERD diagram and the schema developed, showing a subset of the dataset from cleaned 27000 rows to 200 rows. Two hundred rows of data allowed me to show an implementation of the data warehouse based on the project. The only ETL useful found in our university system is SQL loader, but it's outdated. SQL loader achieved the result of transfer the data from the CSV file into our staging table. The results showed different types would be scored differently depending on how the data has being cleaned and the size of the data. Logistic regression or random forest algorithm was identified in the literature review scored highest on similar review data set done by other people. The project got similar results to what has being identified in the literature review related work section. Future work would be using advanced data mining techonlogies such as cloud-like Azure or AWS to run sentiment analysis.

10. References

Bhatt, A., Patel, A., Chheda, H. and Gawande, K., 2015. Amazon review classification and sentiment analysis. *International Journal of Computer Science and Information Technologies*, *6*(6), pp.5107-5110.

Chen, W., Lin, C. and Tai, Y.S., Text-Based Rating Predictions on Amazon Health & Personal Care Product Review.

Elli, M.S. and Wang, Y.F., 2016. Amazon Reviews, business analytics with sentiment analysis.

Fuel, F. (2020) No online customer reviews means BIG problems in 2017 - Fan and Fuel, *Fan and Fuel*, [online] Available at: https://fanandfuel.com/no-online-customer-reviews-means-big-problems-2017/ (Accessed 4 March 2020).

How Online Reviews Influence Sales, (2017) Spiegel Research Centre, [online] Available at: https://spiegel.medill.northwestern.edu/_pdf/Spiegel_Online%20Review_eBook_Jun2017_FINAL.pdf (Accessed 4 March 2020).

Kaggle.com. 2020. *Amazon Cell Phones Reviews*. [online] Available at: <https://www.kaggle.com/grikomsn/amazon-cell-phones-reviews> [Accessed 21 April 2020].

Nasr, F.M.M., Shaaban, S.E.M. and Hafez, T.A.M., 2017. Building Sentiment analysis Model using Graphlab. *International Journal of Scientific and Engineering Research*, *8*, pp.1155-1160.

Nibras, G. (2020) Amazon Cell Phones Reviews, *Kaggle.com*, [online] Available at: https://www.kaggle.com/grikomsn/amazon-cell-phones-reviews/kernels (Accessed 4 March 2020).

Rain, C., 2013. Sentiment analysis in amazon reviews using probabilistic machine learning. *Swarthmore College*.

Shaikh, T. and Deshpande, D., 2016. Feature selection methods in sentiment analysis and sentiment classification of amazon product reviews. *Int J Comput Trends Technol*, *36*(4), pp.225-230.

Xu, Y., Wu, X. and Wang, Q., 2014. Sentiment Analysis of Yelp‘s Ratings Based on Text Reviews.

11. Self-Assessments

Mohamed AL_Kaisi

(Place a tick in the box that you deem to be most indicative of the quality of the work)

	%	No Attempt To very Poor	Poor	Fair	Good	Very good	Excellent
Database design and implementation	20						✓
Data mining	20					✓	
Visualisation	15						✓
Integration	5					✓	
Report> report layout and organization	5				✓		
Report> literature review	7					✓	
Report> discussion and evaluation	8				✓		
Data	5					✓	
Demo as self-recorded video	10						✓
Accurate self-assessment	5					✓	

Totals	**100**						

Note: You must submit this self-assessment as part of the final report (attach it as the last page of your report). The boxes in bold are for examiner use only.