CMPE 257 - PROJECT PROPOSAL

Sentiment Analysis on Product Reviews

Dataset:
https://data.world/datafiniti/amazon-and-best-buy-electronics

GitHub Repository:
https://github.com/Tejasree-Goli/CMPE-257-Project.git

Google Colab [.ipynb]:
https://colab.research.google.com/drive/17iAiIujSmmJuZwQQZOBg-jefQna6B18g?usp=sharing

Team Details [Team 1]:
Pranjali Seth - 015962466
Sai Teja Kandukuri - 016709732
Pavan Satyam - 016422172
Teja Sree Goli - 016040986

Project Title: Sentiment Analysis on Product Reviews

Dataset Source: **https://data.world/datafiniti/amazon-and-best-buy-electronics**

The above data set is taken from Data World [primary source: Datafiniti's Product Database]. The dataset contains around 7200 online reviews posted on e-commerce websites like Amazon, BestBuy and Walmart for various brand products. The data set reviews about 50 electronic products that contain 27 different attributes including reviews title, reviews text, reviews username, reviews rating, product name, manufacturer, brand, image URLs etc.

Problem Statement:

The world is drastically shifting towards the era of online shopping and social media. People find it extremely feasible and less time-consuming to shop online by just sitting and shopping for anything and everything they need from the comfort of their homes. This leads to minimal customer-manufacturer interaction and for this reason, it raises a concern for the suppliers to figure out their product performance and analyze feedback. A manufacturer requires constant feedback on how their products are doing in the market and the level of customer satisfaction that they are delivering.

Therefore, to address this, we have a need for text and sentiment analysis of consumer feedback and product reviews that are purchased by consumers on online platforms. This approach will help in categorizing data based on certain attributes which will make it easier to analyze and observe the trends/reviews of products.

Project Idea:

1. **Objective:** To determine whether a review on a given product is positive or negative by analyzing the text in user reviews on various products and performing a binary classification of each product's reviews.

2. **Approach:** We will be implementing our model on supervised learning methods using word embeddings to predict or classify different sentiments. We plan on experimenting and exploring the data using KNN, SVMs, Random Forest Classifier or different BERT architectures.

   a. **Data Cleaning and Preprocessing:** The raw data is cleaned by removing the rows which have null values for the columns: reviews.rating, reviews.title and reviews.text columns. The duplicate records in the dataset have been dropped. Stopwords have been removed using the Natural Language Toolkit (NLTK) module.

b. **Initial Findings:** Post data visualization, we observed the frequently used words by consumers in the reviews, the frequency of the ratings, the average rating of various brands and the correlation between variables in the data. The reviews are mostly positive, even on the brand/manufacturer level. Moreover, the text reviews are significantly higher for positive ratings and most of the products are recommended by the users which shows an incline towards higher positive ratings in the data.

c. **Challenges:** There is an imbalance in the dataset with a majority of positive ratings. Also, for the negative ratings, the review texts are minimal.

CMPE_257_project_proposal

November 2, 2022

1 Sentiment Analysis on Product Reviews.

Mounting drive for the colab notebook

[1]:
```python
#Mounting the drive for the colab notebook
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

Importing the required libraries

[2]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.tools as tls
import plotly.offline as py
import plotly.graph_objs as go
import warnings

# NLP modules
import nltk
import re
import string
from nltk.corpus import stopwords
from stop_words import get_stop_words
from nltk.stem.porter import PorterStemmer
from textblob import TextBlob , Word
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize

# Wordcloud Modules
from wordcloud import WordCloud , STOPWORDS
```

[3]:
```python
color = sns.color_palette()
warnings.filterwarnings('ignore')
py.init_notebook_mode(connected=True)
```

```python
nltk.download("stopwords")
nltk.download("all")
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading collection 'all'
[nltk_data]    |
[nltk_data]    | Downloading package abc to /root/nltk_data…
[nltk_data]    |   Package abc is already up-to-date!
[nltk_data]    | Downloading package alpino to /root/nltk_data…
[nltk_data]    |   Package alpino is already up-to-date!
[nltk_data]    | Downloading package averaged_perceptron_tagger to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package averaged_perceptron_tagger is already up-
[nltk_data]    |       to-date!
[nltk_data]    | Downloading package averaged_perceptron_tagger_ru to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package averaged_perceptron_tagger_ru is already
[nltk_data]    |       up-to-date!
[nltk_data]    | Downloading package basque_grammars to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package basque_grammars is already up-to-date!
[nltk_data]    | Downloading package biocreative_ppi to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package biocreative_ppi is already up-to-date!
[nltk_data]    | Downloading package bllip_wsj_no_aux to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package bllip_wsj_no_aux is already up-to-date!
[nltk_data]    | Downloading package book_grammars to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package book_grammars is already up-to-date!
[nltk_data]    | Downloading package brown to /root/nltk_data…
[nltk_data]    |   Package brown is already up-to-date!
[nltk_data]    | Downloading package brown_tei to /root/nltk_data…
[nltk_data]    |   Package brown_tei is already up-to-date!
[nltk_data]    | Downloading package cess_cat to /root/nltk_data…
[nltk_data]    |   Package cess_cat is already up-to-date!
[nltk_data]    | Downloading package cess_esp to /root/nltk_data…
[nltk_data]    |   Package cess_esp is already up-to-date!
[nltk_data]    | Downloading package chat80 to /root/nltk_data…
[nltk_data]    |   Package chat80 is already up-to-date!
[nltk_data]    | Downloading package city_database to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package city_database is already up-to-date!
[nltk_data]    | Downloading package cmudict to /root/nltk_data…
[nltk_data]    |   Package cmudict is already up-to-date!
[nltk_data]    | Downloading package comparative_sentences to
```

```
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package comparative_sentences is already up-to-
[nltk_data]    |       date!
[nltk_data]    | Downloading package comtrans to /root/nltk_data…
[nltk_data]    |   Package comtrans is already up-to-date!
[nltk_data]    | Downloading package conll2000 to /root/nltk_data…
[nltk_data]    |   Package conll2000 is already up-to-date!
[nltk_data]    | Downloading package conll2002 to /root/nltk_data…
[nltk_data]    |   Package conll2002 is already up-to-date!
[nltk_data]    | Downloading package conll2007 to /root/nltk_data…
[nltk_data]    |   Package conll2007 is already up-to-date!
[nltk_data]    | Downloading package crubadan to /root/nltk_data…
[nltk_data]    |   Package crubadan is already up-to-date!
[nltk_data]    | Downloading package dependency_treebank to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package dependency_treebank is already up-to-date!
[nltk_data]    | Downloading package dolch to /root/nltk_data…
[nltk_data]    |   Package dolch is already up-to-date!
[nltk_data]    | Downloading package europarl_raw to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package europarl_raw is already up-to-date!
[nltk_data]    | Downloading package extended_omw to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package extended_omw is already up-to-date!
[nltk_data]    | Downloading package floresta to /root/nltk_data…
[nltk_data]    |   Package floresta is already up-to-date!
[nltk_data]    | Downloading package framenet_v15 to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package framenet_v15 is already up-to-date!
[nltk_data]    | Downloading package framenet_v17 to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package framenet_v17 is already up-to-date!
[nltk_data]    | Downloading package gazetteers to /root/nltk_data…
[nltk_data]    |   Package gazetteers is already up-to-date!
[nltk_data]    | Downloading package genesis to /root/nltk_data…
[nltk_data]    |   Package genesis is already up-to-date!
[nltk_data]    | Downloading package gutenberg to /root/nltk_data…
[nltk_data]    |   Package gutenberg is already up-to-date!
[nltk_data]    | Downloading package ieer to /root/nltk_data…
[nltk_data]    |   Package ieer is already up-to-date!
[nltk_data]    | Downloading package inaugural to /root/nltk_data…
[nltk_data]    |   Package inaugural is already up-to-date!
[nltk_data]    | Downloading package indian to /root/nltk_data…
[nltk_data]    |   Package indian is already up-to-date!
[nltk_data]    | Downloading package jeita to /root/nltk_data…
[nltk_data]    |   Package jeita is already up-to-date!
[nltk_data]    | Downloading package kimmo to /root/nltk_data…
[nltk_data]    |   Package kimmo is already up-to-date!
```

```
[nltk_data]    | Downloading package knbc to /root/nltk_data…
[nltk_data]    |   Package knbc is already up-to-date!
[nltk_data]    | Downloading package large_grammars to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package large_grammars is already up-to-date!
[nltk_data]    | Downloading package lin_thesaurus to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package lin_thesaurus is already up-to-date!
[nltk_data]    | Downloading package mac_morpho to /root/nltk_data…
[nltk_data]    |   Package mac_morpho is already up-to-date!
[nltk_data]    | Downloading package machado to /root/nltk_data…
[nltk_data]    |   Package machado is already up-to-date!
[nltk_data]    | Downloading package masc_tagged to /root/nltk_data…
[nltk_data]    |   Package masc_tagged is already up-to-date!
[nltk_data]    | Downloading package maxent_ne_chunker to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package maxent_ne_chunker is already up-to-date!
[nltk_data]    | Downloading package maxent_treebank_pos_tagger to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package maxent_treebank_pos_tagger is already up-
[nltk_data]    |       to-date!
[nltk_data]    | Downloading package moses_sample to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package moses_sample is already up-to-date!
[nltk_data]    | Downloading package movie_reviews to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package movie_reviews is already up-to-date!
[nltk_data]    | Downloading package mte_teip5 to /root/nltk_data…
[nltk_data]    |   Package mte_teip5 is already up-to-date!
[nltk_data]    | Downloading package mwa_ppdb to /root/nltk_data…
[nltk_data]    |   Package mwa_ppdb is already up-to-date!
[nltk_data]    | Downloading package names to /root/nltk_data…
[nltk_data]    |   Package names is already up-to-date!
[nltk_data]    | Downloading package nombank.1.0 to /root/nltk_data…
[nltk_data]    |   Package nombank.1.0 is already up-to-date!
[nltk_data]    | Downloading package nonbreaking_prefixes to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package nonbreaking_prefixes is already up-to-date!
[nltk_data]    | Downloading package nps_chat to /root/nltk_data…
[nltk_data]    |   Package nps_chat is already up-to-date!
[nltk_data]    | Downloading package omw to /root/nltk_data…
[nltk_data]    |   Package omw is already up-to-date!
[nltk_data]    | Downloading package omw-1.4 to /root/nltk_data…
[nltk_data]    |   Package omw-1.4 is already up-to-date!
[nltk_data]    | Downloading package opinion_lexicon to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package opinion_lexicon is already up-to-date!
[nltk_data]    | Downloading package panlex_swadesh to
```

```
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package panlex_swadesh is already up-to-date!
[nltk_data]    | Downloading package paradigms to /root/nltk_data…
[nltk_data]    |   Package paradigms is already up-to-date!
[nltk_data]    | Downloading package pe08 to /root/nltk_data…
[nltk_data]    |   Package pe08 is already up-to-date!
[nltk_data]    | Downloading package perluniprops to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package perluniprops is already up-to-date!
[nltk_data]    | Downloading package pil to /root/nltk_data…
[nltk_data]    |   Package pil is already up-to-date!
[nltk_data]    | Downloading package pl196x to /root/nltk_data…
[nltk_data]    |   Package pl196x is already up-to-date!
[nltk_data]    | Downloading package porter_test to /root/nltk_data…
[nltk_data]    |   Package porter_test is already up-to-date!
[nltk_data]    | Downloading package ppattach to /root/nltk_data…
[nltk_data]    |   Package ppattach is already up-to-date!
[nltk_data]    | Downloading package problem_reports to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package problem_reports is already up-to-date!
[nltk_data]    | Downloading package product_reviews_1 to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package product_reviews_1 is already up-to-date!
[nltk_data]    | Downloading package product_reviews_2 to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package product_reviews_2 is already up-to-date!
[nltk_data]    | Downloading package propbank to /root/nltk_data…
[nltk_data]    |   Package propbank is already up-to-date!
[nltk_data]    | Downloading package pros_cons to /root/nltk_data…
[nltk_data]    |   Package pros_cons is already up-to-date!
[nltk_data]    | Downloading package ptb to /root/nltk_data…
[nltk_data]    |   Package ptb is already up-to-date!
[nltk_data]    | Downloading package punkt to /root/nltk_data…
[nltk_data]    |   Package punkt is already up-to-date!
[nltk_data]    | Downloading package qc to /root/nltk_data…
[nltk_data]    |   Package qc is already up-to-date!
[nltk_data]    | Downloading package reuters to /root/nltk_data…
[nltk_data]    |   Package reuters is already up-to-date!
[nltk_data]    | Downloading package rslp to /root/nltk_data…
[nltk_data]    |   Package rslp is already up-to-date!
[nltk_data]    | Downloading package rte to /root/nltk_data…
[nltk_data]    |   Package rte is already up-to-date!
[nltk_data]    | Downloading package sample_grammars to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package sample_grammars is already up-to-date!
[nltk_data]    | Downloading package semcor to /root/nltk_data…
[nltk_data]    |   Package semcor is already up-to-date!
[nltk_data]    | Downloading package senseval to /root/nltk_data…
```

```
[nltk_data]    |   Package senseval is already up-to-date!
[nltk_data]    | Downloading package sentence_polarity to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package sentence_polarity is already up-to-date!
[nltk_data]    | Downloading package sentiwordnet to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package sentiwordnet is already up-to-date!
[nltk_data]    | Downloading package shakespeare to /root/nltk_data…
[nltk_data]    |   Package shakespeare is already up-to-date!
[nltk_data]    | Downloading package sinica_treebank to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package sinica_treebank is already up-to-date!
[nltk_data]    | Downloading package smultron to /root/nltk_data…
[nltk_data]    |   Package smultron is already up-to-date!
[nltk_data]    | Downloading package snowball_data to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package snowball_data is already up-to-date!
[nltk_data]    | Downloading package spanish_grammars to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package spanish_grammars is already up-to-date!
[nltk_data]    | Downloading package state_union to /root/nltk_data…
[nltk_data]    |   Package state_union is already up-to-date!
[nltk_data]    | Downloading package stopwords to /root/nltk_data…
[nltk_data]    |   Package stopwords is already up-to-date!
[nltk_data]    | Downloading package subjectivity to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package subjectivity is already up-to-date!
[nltk_data]    | Downloading package swadesh to /root/nltk_data…
[nltk_data]    |   Package swadesh is already up-to-date!
[nltk_data]    | Downloading package switchboard to /root/nltk_data…
[nltk_data]    |   Package switchboard is already up-to-date!
[nltk_data]    | Downloading package tagsets to /root/nltk_data…
[nltk_data]    |   Package tagsets is already up-to-date!
[nltk_data]    | Downloading package timit to /root/nltk_data…
[nltk_data]    |   Package timit is already up-to-date!
[nltk_data]    | Downloading package toolbox to /root/nltk_data…
[nltk_data]    |   Package toolbox is already up-to-date!
[nltk_data]    | Downloading package treebank to /root/nltk_data…
[nltk_data]    |   Package treebank is already up-to-date!
[nltk_data]    | Downloading package twitter_samples to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package twitter_samples is already up-to-date!
[nltk_data]    | Downloading package udhr to /root/nltk_data…
[nltk_data]    |   Package udhr is already up-to-date!
[nltk_data]    | Downloading package udhr2 to /root/nltk_data…
[nltk_data]    |   Package udhr2 is already up-to-date!
[nltk_data]    | Downloading package unicode_samples to
[nltk_data]    |     /root/nltk_data…
```

```
[nltk_data]    |   Package unicode_samples is already up-to-date!
[nltk_data]    | Downloading package universal_tagset to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package universal_tagset is already up-to-date!
[nltk_data]    | Downloading package universal_treebanks_v20 to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package universal_treebanks_v20 is already up-to-
[nltk_data]    |       date!
[nltk_data]    | Downloading package vader_lexicon to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package vader_lexicon is already up-to-date!
[nltk_data]    | Downloading package verbnet to /root/nltk_data…
[nltk_data]    |   Package verbnet is already up-to-date!
[nltk_data]    | Downloading package verbnet3 to /root/nltk_data…
[nltk_data]    |   Package verbnet3 is already up-to-date!
[nltk_data]    | Downloading package webtext to /root/nltk_data…
[nltk_data]    |   Package webtext is already up-to-date!
[nltk_data]    | Downloading package wmt15_eval to /root/nltk_data…
[nltk_data]    |   Package wmt15_eval is already up-to-date!
[nltk_data]    | Downloading package word2vec_sample to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package word2vec_sample is already up-to-date!
[nltk_data]    | Downloading package wordnet to /root/nltk_data…
[nltk_data]    |   Package wordnet is already up-to-date!
[nltk_data]    | Downloading package wordnet2021 to /root/nltk_data…
[nltk_data]    |   Package wordnet2021 is already up-to-date!
[nltk_data]    | Downloading package wordnet31 to /root/nltk_data…
[nltk_data]    |   Package wordnet31 is already up-to-date!
[nltk_data]    | Downloading package wordnet_ic to /root/nltk_data…
[nltk_data]    |   Package wordnet_ic is already up-to-date!
[nltk_data]    | Downloading package words to /root/nltk_data…
[nltk_data]    |   Package words is already up-to-date!
[nltk_data]    | Downloading package ycoe to /root/nltk_data…
[nltk_data]    |   Package ycoe is already up-to-date!
[nltk_data]    |
[nltk_data]  Done downloading collection all
```

```
[3]: True
```

1.1 Understanding data

Load/Read the dataset

```
[4]: reviews_df=pd.read_csv('/content/drive/MyDrive/amazon_dataset/product.csv')
     reviews_df.head(5)
```

```
[4]:                      id       asins      brand  \
     0  AVpf3txeLJeJML43FN82  B0168YIWSI  Microsoft
     1  AVpf3txeLJeJML43FN82  B0168YIWSI  Microsoft
     2  AVpf3txeLJeJML43FN82  B0168YIWSI  Microsoft
     3  AVpf3txeLJeJML43FN82  B0168YIWSI  Microsoft
     4  AVpf3txeLJeJML43FN82  B0168YIWSI  Microsoft

                                               categories colors  \
     0  Electronics,Computers,Computer Accessories,Key…  Black
     1  Electronics,Computers,Computer Accessories,Key…  Black
     2  Electronics,Computers,Computer Accessories,Key…  Black
     3  Electronics,Computers,Computer Accessories,Key…  Black
     4  Electronics,Computers,Computer Accessories,Key…  Black

                   dateAdded           dateUpdated                   dimension  \
     0  2015-11-13T12:28:09Z  2018-01-29T02:15:13Z  11.6 in x 8.5 in x 0.19 in
     1  2015-11-13T12:28:09Z  2018-01-29T02:15:13Z  11.6 in x 8.5 in x 0.19 in
     2  2015-11-13T12:28:09Z  2018-01-29T02:15:13Z  11.6 in x 8.5 in x 0.19 in
     3  2015-11-13T12:28:09Z  2018-01-29T02:15:13Z  11.6 in x 8.5 in x 0.19 in
     4  2015-11-13T12:28:09Z  2018-01-29T02:15:13Z  11.6 in x 8.5 in x 0.19 in

                 ean                                        imageURLs  …  \
     0  8.900000e+11  https://i5.walmartimages.com/asr/2a41f6f0-844e…  …
     1  8.900000e+11  https://i5.walmartimages.com/asr/2a41f6f0-844e…  …
     2  8.900000e+11  https://i5.walmartimages.com/asr/2a41f6f0-844e…  …
     3  8.900000e+11  https://i5.walmartimages.com/asr/2a41f6f0-844e…  …
     4  8.900000e+11  https://i5.walmartimages.com/asr/2a41f6f0-844e…  …

       reviews.doRecommend reviews.numHelpful reviews.rating  \
     0                True                0.0            5.0
     1                True                0.0            4.0
     2                True                0.0            4.0
     3                True                0.0            5.0
     4                True                0.0            5.0

                                      reviews.sourceURLs  \
     0  http://reviews.bestbuy.com/3545/4562009/review…
     1  http://reviews.bestbuy.com/3545/4562009/review…
     2  http://reviews.bestbuy.com/3545/4562009/review…
     3  http://reviews.bestbuy.com/3545/4562009/review…
     4  http://reviews.bestbuy.com/3545/4562009/review…

                                            reviews.text  \
     0  This keyboard is very easy to type on, but the…
     1  It's thin and light. I can type pretty easily …
     2  I love the new design the keys are spaced well…
     3  Attached easily and firmly. Has a nice feel. A…
```

```
4  Our original keyboard was okay, but did not ha…

                 reviews.title reviews.username  \
0  Love the fingerprint reader             JNH1
1                         Nice             Appa
2                          New             Kman
3                 Nice keyboard        UpstateNY
4              Nice improvement        Glickster

                                          sourceURLs           upc      weight
0  https://www.walmart.com/ip/Microsoft-Surface-P…  8.900000e+11  1.1 pounds
1  https://www.walmart.com/ip/Microsoft-Surface-P…  8.900000e+11  1.1 pounds
2  https://www.walmart.com/ip/Microsoft-Surface-P…  8.900000e+11  1.1 pounds
3  https://www.walmart.com/ip/Microsoft-Surface-P…  8.900000e+11  1.1 pounds
4  https://www.walmart.com/ip/Microsoft-Surface-P…  8.900000e+11  1.1 pounds

[5 rows x 27 columns]
```

Shape of the dataframe

[5]:
```
reviews_df.shape
```

[5]:
```
(7299, 27)
```

There are 27 columns and a total of 7299 rows in this dataset.

[6]:
```
#Columns/attributes and their datatypes
reviews_df.dtypes
```

[6]:
```
id                        object
asins                     object
brand                     object
categories                object
colors                    object
dateAdded                 object
dateUpdated               object
dimension                 object
ean                      float64
imageURLs                 object
keys                      object
manufacturer              object
manufacturerNumber        object
name                      object
primaryCategories         object
reviews.date              object
reviews.dateSeen          object
reviews.doRecommend       object
reviews.numHelpful       float64
reviews.rating         float64
reviews.sourceURLs      object
reviews.text            object
reviews.title           object
reviews.username        object
sourceURLs              object
upc                    float64
weight                  object
dtype: object
```

The columns reflect on different attributes that are useful in understanding the reviews on products. We mainly look at the brand manufacturers, recommendations, ratings, and user reviews for different products sold on Amazon, Ebay, etc.

1.2 Data Cleaning and preprocessing

[7]:

```
reviews_df.isnull().sum()
```

[7]:

```
id                        0
asins                     0
brand                     0
categories                0
colors                 2019
dateAdded                 0
dateUpdated               0
dimension              1209
ean                    4348
imageURLs                 0
keys                      0
manufacturer           2667
manufacturerNumber        0
name                      0
primaryCategories         0
reviews.date             61
reviews.dateSeen          0
reviews.doRecommend    1391
reviews.numHelpful     1486
reviews.rating          164
reviews.sourceURLs        0
reviews.text              5
reviews.title             4
reviews.username          0
sourceURLs                0
upc                       0
weight                    0
dtype: int64
```

We look at the null values in the data to drop them. The fields that are most used for sentiment classification in the data are user review in text and the rating of the product. All the null values are dropped.

```
[8]: reviews_df = reviews_df.dropna(subset=['reviews.text']) #dropping null reviews
     reviews_df = reviews_df.dropna(subset=['reviews.rating']) #dropping null ratings
```

```
[9]: reviews_df.shape
```

```
[9]: (7130, 27)
```

Then we get rid of the duplicate values in the text. We match the text of the review, rating, username and the date when the review was posted to identify the duplicate values and drop them.

```
[10]: reviews_df.duplicated(subset=['reviews.text', 'reviews.username', 'reviews.
        ↪rating', 'reviews.date']).sum()
```

```
[10]: 14
```

```
[11]: reviews_df=reviews_df.drop_duplicates(subset=['reviews.text', 'reviews.
        ↪username', 'reviews.rating', 'reviews.date'])
```

```
[12]: reviews_df.shape
```

```
[12]: (7116, 27)
```

After dropping null values and duplicate entries, there are now 7116 rows in the data.

We then convert our reviews to all lowercase text and remove the unnecessary string literals from the text for proper preprocessing. This is done to avoid having different representations of the same word in the vector space. We remove the stopwords in order to remove the low level information from our text and give more focus to the important information.

```
[13]: reviews_df["reviews.text"] = (
          reviews_df["reviews.text"]
          .str.lower()
          .str.replace("[^\w\s]", "")
          .str.replace("\d+", "")
          .str.replace("\n", " ")
          .replace("\r", "")
          .str.replace("[^a-zA-Z0-9\s]", "")
      )
```

```
[14]: def word_cleaner(data):
          words = [re.sub("[^a-zA-Z]", " ", i) for i in data]
          words = [i.lower() for j in words for i in j.split()] # Split all the␣
        ↪sentences into words
          words = [i for i in words if not i in set(stopwords.words("english"))] #␣
        ↪Split all the sentences into words
```

```
    return words
```

We identify the most common used words in the text to analyze them in product reviews and plot the frequency of these words. The words such as “great” and “remote” are used frequently in the reviews.

[15]:

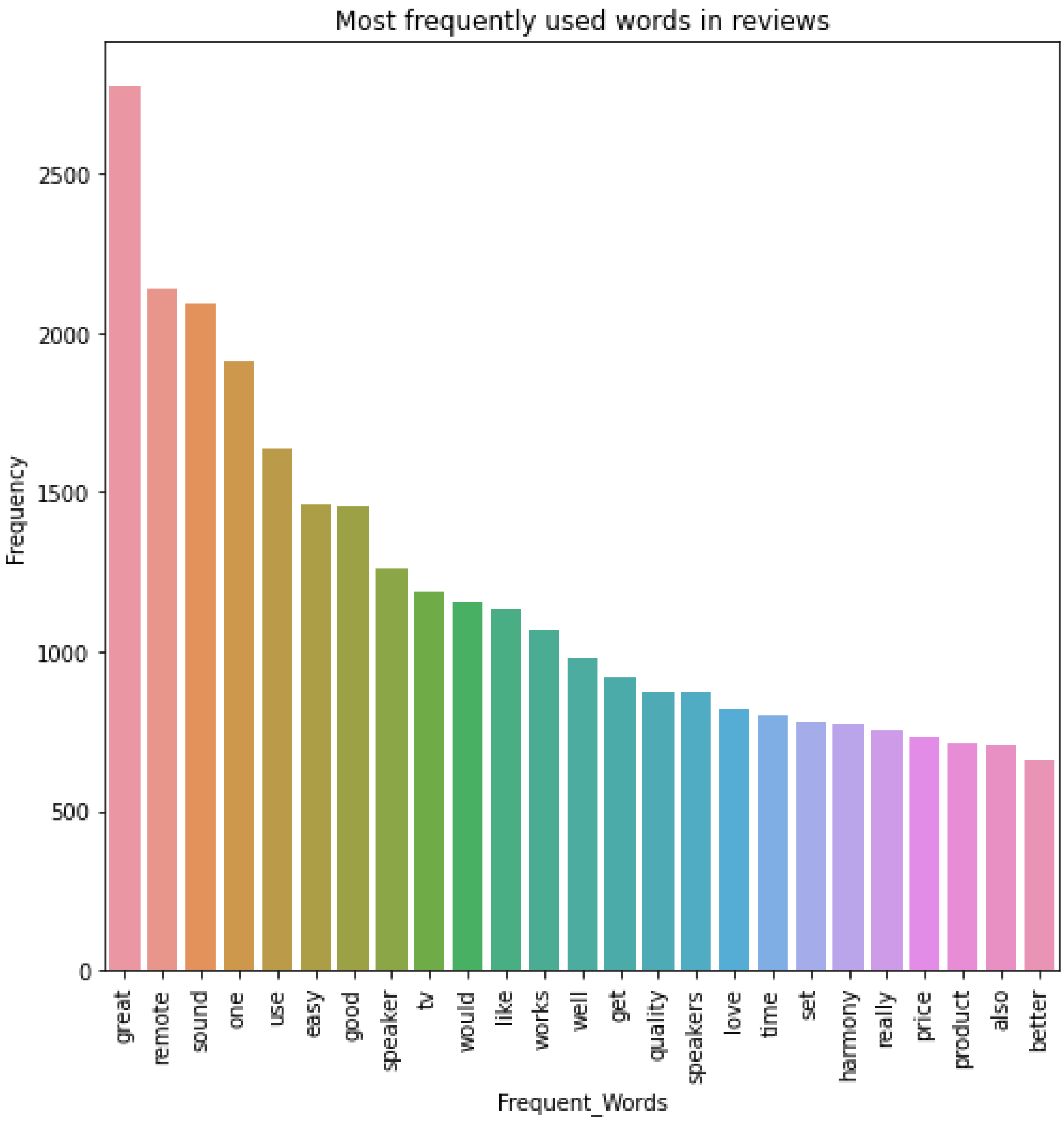
```
word_frequency = pd.DataFrame(
    nltk.FreqDist(word_cleaner(reviews_df["reviews.text"])).most_common(25),
    columns=["Frequent_Words", "Frequency"],
)
```

[16]:

```
plt.figure(figsize=(8, 8))
plt.xticks(rotation=90)
plt.title("Most frequently used words in reviews")
sns.barplot(x="Frequent_Words", y="Frequency", data=word_frequency)
```

[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3359c75c90>
```



```
[17]: lemmatizer_output = WordNetLemmatizer()

      reviews_df["reviews.text"] = reviews_df["reviews.text"].apply(
          lambda x: word_tokenize(x.lower())
      )
      reviews_df["reviews.text"] = reviews_df["reviews.text"].apply(
          lambda x: [word for word in x if word not in STOPWORDS]
      )
      reviews_df["reviews.text"] = reviews_df["reviews.text"].apply(
          lambda x: [lemmatizer_output.lemmatize(word) for word in x]
      )
```

```
reviews_df["reviews.text"] = reviews_df["reviews.text"].apply(lambda x: " ".
 ↪join(x))
```

[18]:

```
reviews_df['reviews.text'].head(10)
```

[18]:

```
0    keyboard easy type fingerprint reader best fea…
1                        thin light type pretty easily
2    love new design key spaced well mi type finger…
3    attached easily firmly nice feel must surface pro
4    original keyboard okay laptop feel bit floppy …
5    purchased replace original surface pro keyboar…
6      find comfortable type rarely use fingerprint id
7    good keyboard addition surface pro platform de…
8    tough getting work surface pro worked bug love…
9    now quickly hassle free log surface finger pri…
Name: reviews.text, dtype: object
```

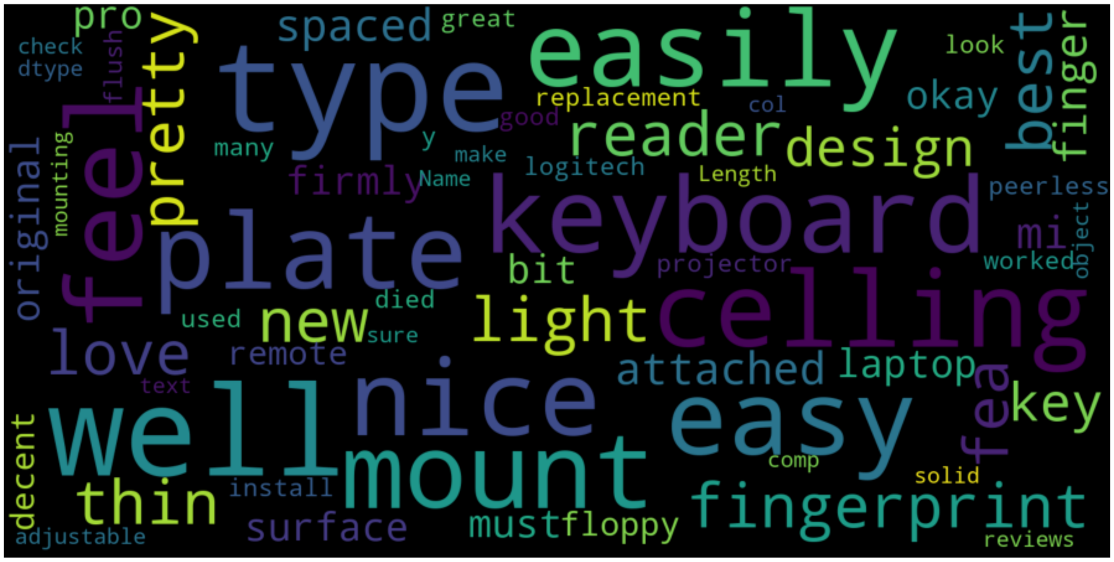
1.3 Visualization

A word cloud can be considered as a snapshot of the text. It is useful in understanding the text at a glance.

[19]:

```
from wordcloud import WordCloud, STOPWORDS

stopwords = set(STOPWORDS)


def show_wordcloud(data, title=None):
    wordcloud = WordCloud(
        background_color="black",
        stopwords=stopwords,
        max_words=250,
        max_font_size=45,
        scale=4,
        random_state=1,
    ).generate(str(data))

    fig = plt.figure(1, figsize=(16, 16))
    plt.axis("off")
    if title:
        fig.suptitle(title, fontsize=21)
        fig.subplots_adjust(top=2.1)

    plt.imshow(wordcloud)
    plt.show()
```

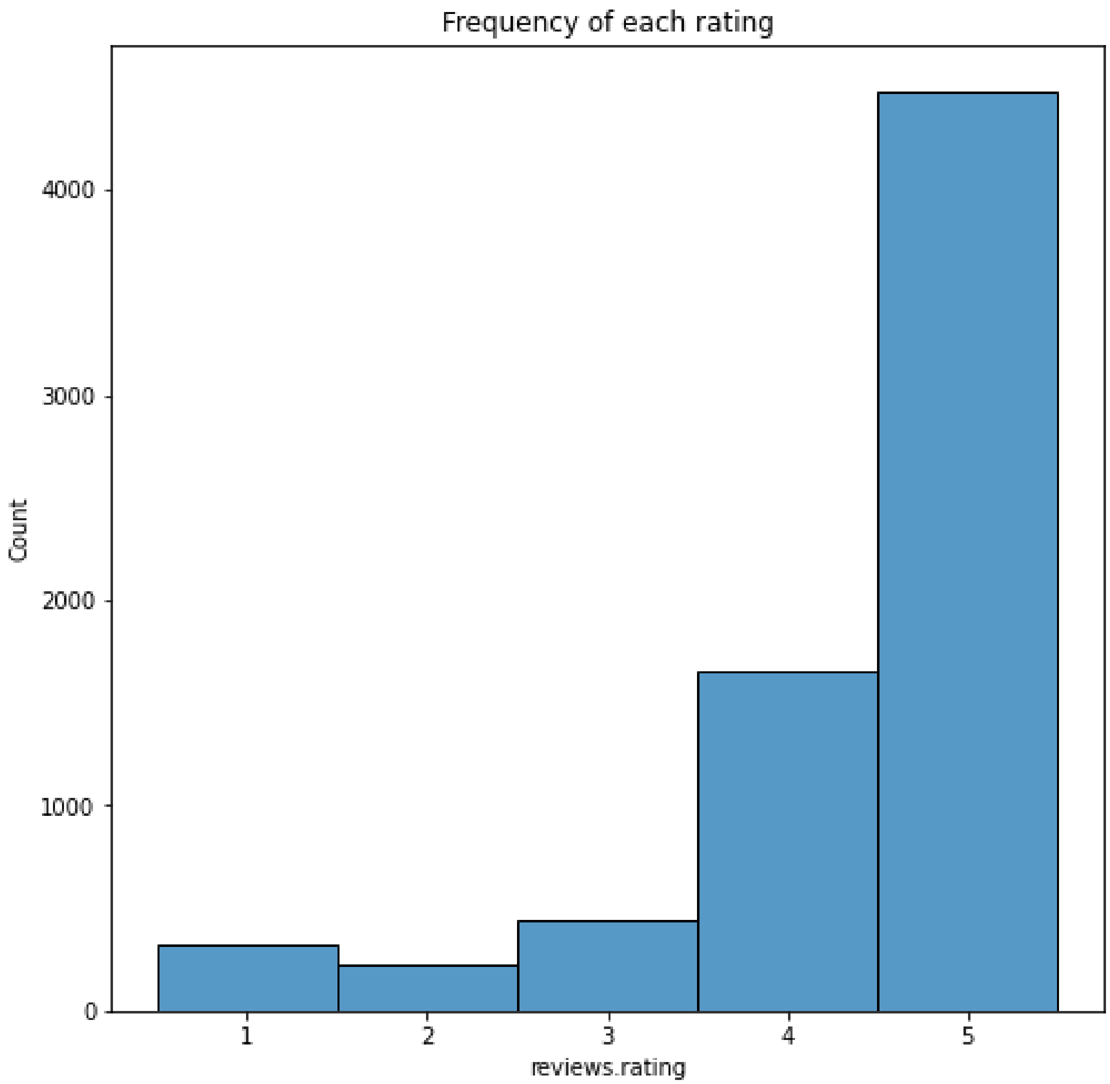
```
show_wordcloud(reviews_df["reviews.text"])
```



Plotting the frequency of ratings from 0 stars to 5 stars.

[20]:
```
plt.figure(figsize=(8,8))
sns.histplot(data=reviews_df, x=reviews_df['reviews.rating'], discrete="True").
 ↪set(title = "Frequency of each rating")
```

[20]:
```
[Text(0.5, 1.0, 'Frequency of each rating')]
```



The distribution here is mostly positive (4 and 5 stars) and implies that the customers are happy with the products they purchase.

We also look at the reviews of each brand. When predicting the sentiment labels for customer satisfaction, this could be useful to understand the customer satisfaction for a particular brand.

```
[21]: #review by brand
      reviews_df.groupby(reviews_df['brand']).mean()['reviews.rating']
```

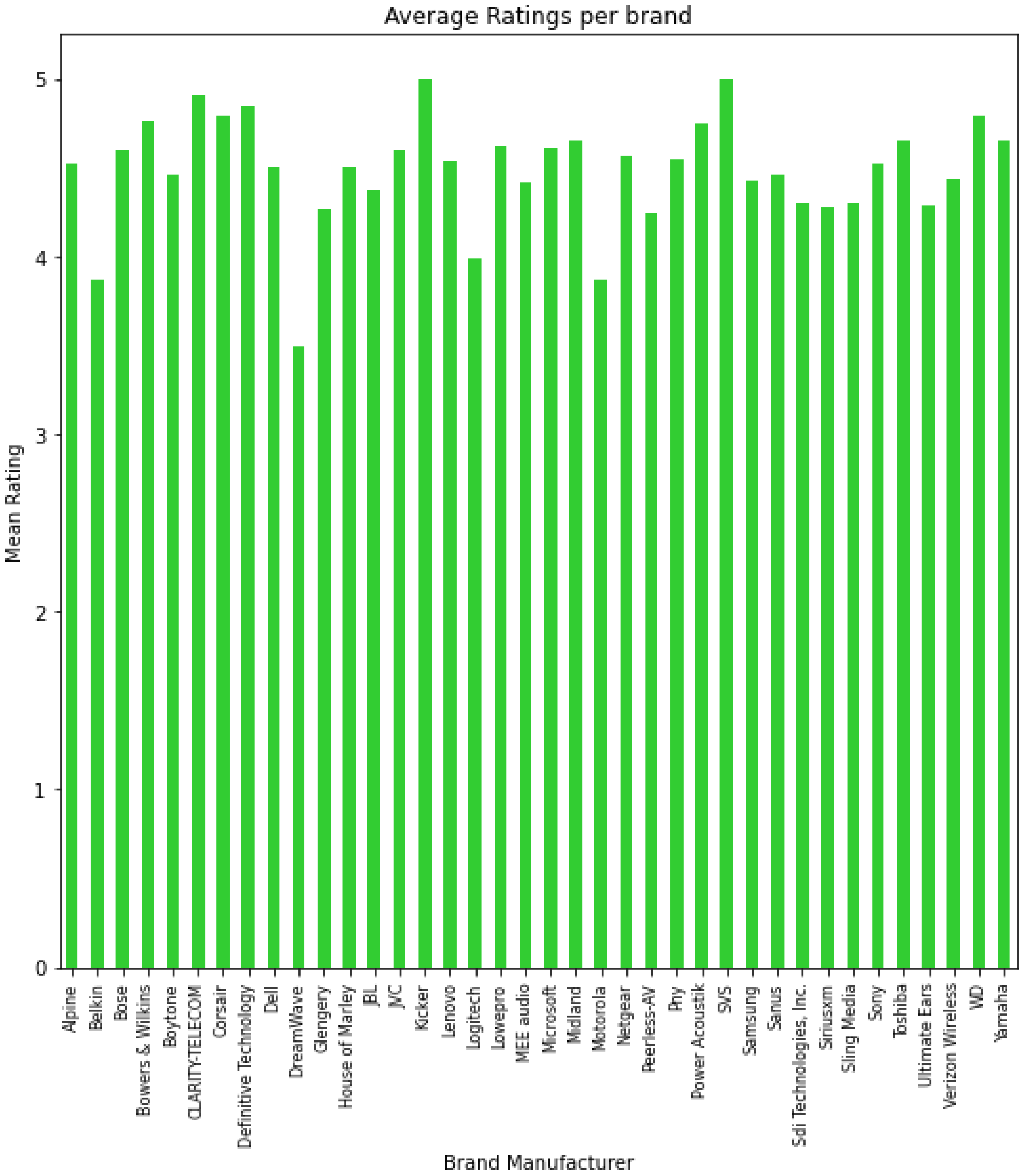
```
[21]: brand
      Alpine                    4.526923
      Belkin                    3.875000
      Bose                      4.600000
      Bowers & Wilkins          4.766355
```

```
Boytone                     4.459459
CLARITY-TELECOM             4.909091
Corsair                     4.798246
Definitive Technology       4.851852
Dell                        4.500000
DreamWave                   3.500000
Glengery                    4.263158
House of Marley             4.500000
JBL                         4.370044
JVC                         4.604478
Kicker                      5.000000
Lenovo                      4.535714
Logitech                    3.992908
Lowepro                     4.625954
MEE audio                   4.412903
Microsoft                   4.606061
Midland                     4.659091
Motorola                    3.868421
Netgear                     4.570470
Peerless-AV                 4.250000
Pny                         4.549738
Power Acoustik              4.750000
SVS                         5.000000
Samsung                     4.423445
Sanus                       4.456790
Sdi Technologies, Inc.      4.298701
Siriusxm                    4.277778
Sling Media                 4.301170
Sony                        4.522705
Toshiba                     4.652174
Ultimate Ears               4.290000
Verizon Wireless            4.435714
WD                          4.796296
Yamaha                      4.657143
Name: reviews.rating, dtype: float64
```

[22]:

```
reviews_df = reviews_df.replace(np.nan, 0)
reviews_dfm = reviews_df.groupby(reviews_df["brand"]).mean()["reviews.rating"]
plt.title("Average Ratings per brand")
plt.xticks(fontsize=8)
reviews_dfm.plot(
    kind="bar",
    ylabel="Mean Rating",
    xlabel="Brand Manufacturer",
    figsize=(9, 9),
    color="limegreen",
)
```

```
[22]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3357e36450>
```

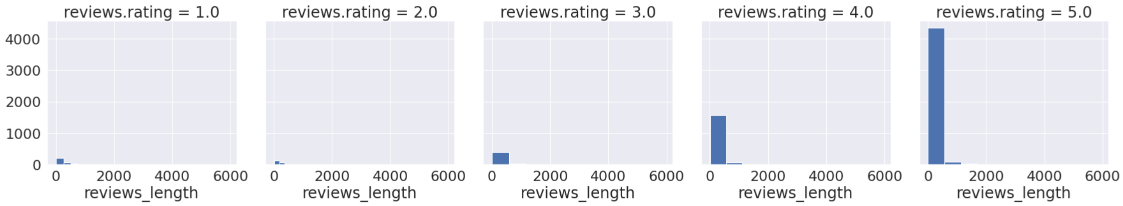


To understand the data, we plot the graphs for length of text in reviews. The users tend to give little or no written review for low ratings. For high ratings, the average review length is about 60 to 80.

```
[23]: reviews_df["reviews_length"] = reviews_df["reviews.text"].apply(len)
      sns.set(font_scale=2.0)
```

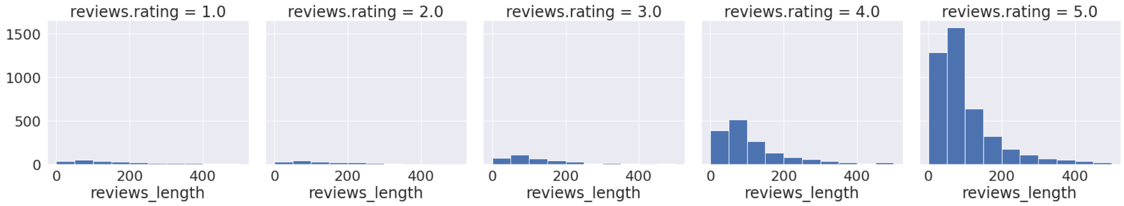
```
graph = sns.FacetGrid(reviews_df, col="reviews.rating", size=5)
graph.map(plt.hist, "reviews_length")
```

```
[23]: <seaborn.axisgrid.FacetGrid at 0x7f3357f23890>
```



```
[24]: graph = sns.FacetGrid(reviews_df,col='reviews.rating',size=5)
      graph.map(plt.hist,'reviews_length', range=[0, 500])
```

```
[24]: <seaborn.axisgrid.FacetGrid at 0x7f3357936890>
```

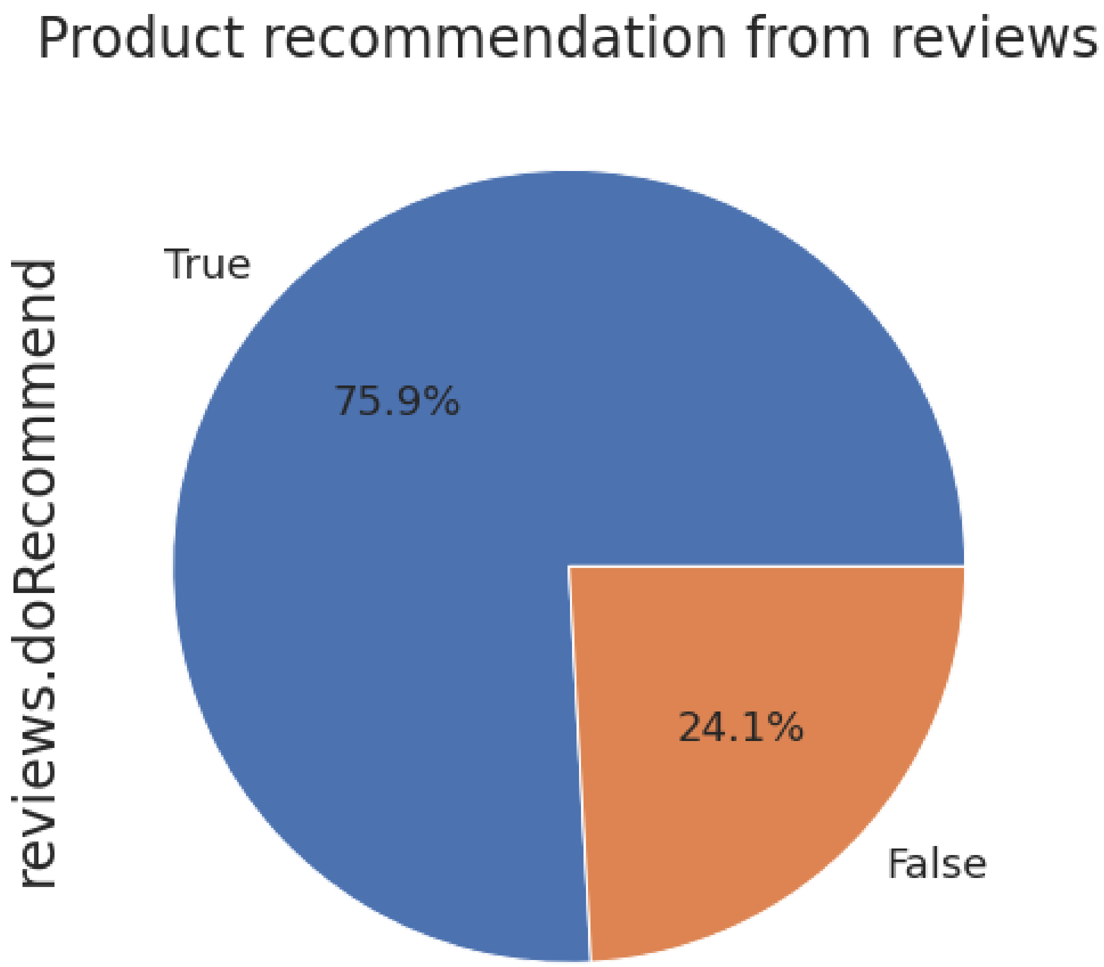


Product recommendation by the users or the e-commerce sites such as Amazon and Ebay also gives information about the customer satisfaction. From the pie plot that is shown below, the recommendations are fairly positive.

```
[25]: reviews_df['reviews.doRecommend'].fillna("N/A",inplace=True)
```

```
[26]: plt.figure(figsize = (8,8))
      plt.title("Product recommendation from reviews")
      reviews_df["reviews.doRecommend"].value_counts().plot.pie(autopct="%1.
       ↪1f%%",textprops={'fontsize': 18})
```

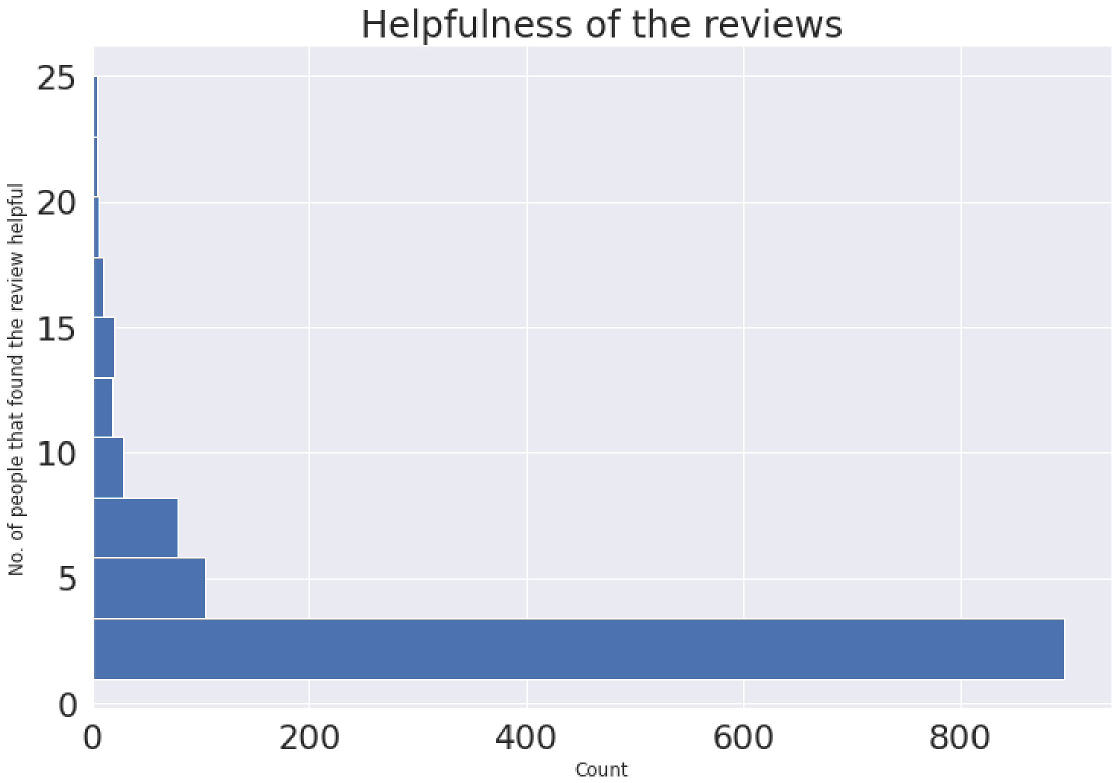
```
[26]: <matplotlib.axes._subplots.AxesSubplot at 0x7f33579a6050>
```



Plotting the count of reviews that are found useful to others when shopping online.

```
[27]: plt.figure(figsize=(12,8))
      plt.hist(reviews_df['reviews.numHelpful'],range=[1, 25],␣
       ↪orientation='horizontal')
      plt.title("Helpfulness of the reviews")
      plt.xlabel("Count", fontsize=12)
      plt.ylabel("No. of people that found the review helpful", fontsize=12)
```

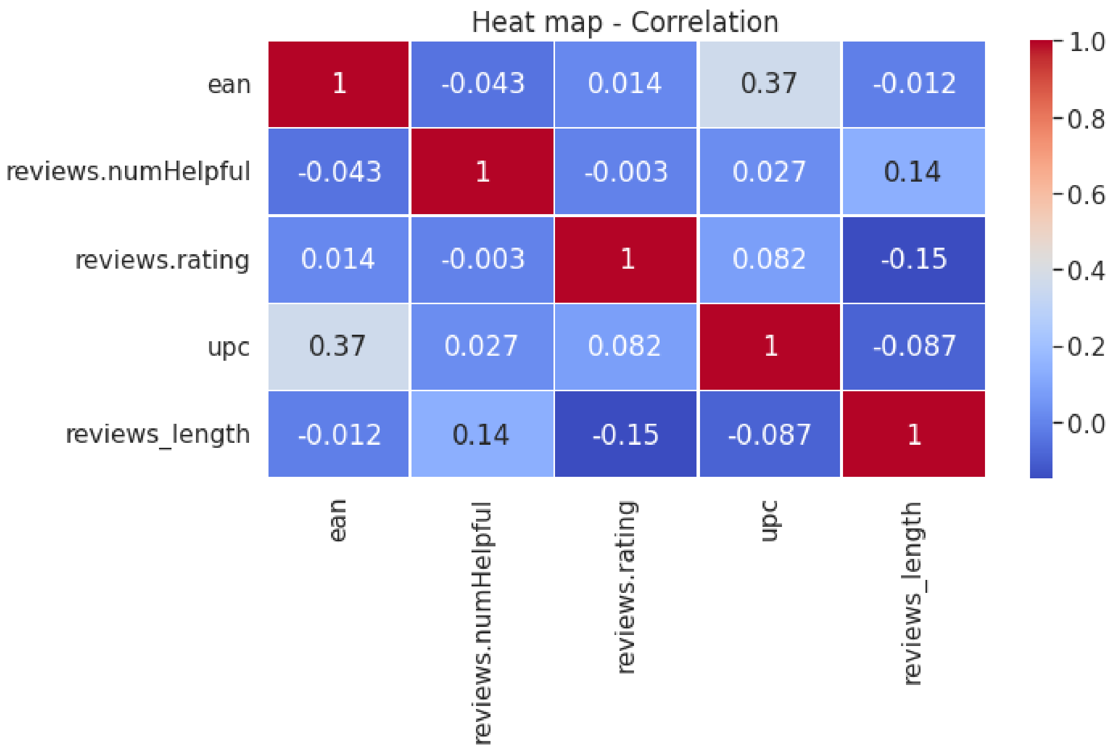
```
[27]: Text(0, 0.5, 'No. of people that found the review helpful')
```



Correlation measures the strength of the relationship between different variables in the data. When a value of one variable changes, it effects the other variable in a certain way.

```
[28]: sns.set(font_scale=1.4)
      plt.figure(figsize = (10,5))
      plt.title("Heat map - Correlation")
      sns.heatmap(reviews_df.corr(),cmap='coolwarm',annot=True,linewidths=.5)
```

```
[28]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3357984690>
```


Heat map - Correlation
ean
reviews.numHelpful
reviews.rating
upc
reviews_length
1
-0.043
0.014
0.37
-0.012
-0.043
1
-0.003
0.027
0.14
0.014
-0.003
1
0.082
-0.15
0.37
0.027
0.082
1
-0.087
-0.012
0.14
-0.15
-0.087
1
ean
reviews.numHelpful
reviews.rating
upc
reviews_length
1.0
0.8
0.6
0.4
0.2
0.0

[ ]: