Amazon Product Reviews Sentiment Analysis using NLP

Authors: ***

- Wambui Githinji
- Lynette Mwiti
- Felix Njoroge
- Wilfred Lekishorumongi
- Monica Mwangi
- Joan Maina

. . .

Problem Statement

Reviews are critical to businesses as they offer insights into customer satisfaction, preferences and areas of improvement.

Businesses need to understand and interpret these reviews in order to cut through the competition. Lots of reviews are generated daily and manually analyzing them is impractical.

Objectives

Use Sentiment analysis to help the businesses get actionable insights from the feedback received from customers.

The approach taken with the analysis seeks to

- Determine the sentiment of the reviews (positive or negative) to understand overall customer satisfaction and feedback.
- Utilize sentiment analysis to help our stakeholders understand customer preferences across various products.
- Conduct exploratory data analysis to understand the distribution of sentiments over time, across barands and products.
- Leverage customer reviews to identify areas for improvement in products based on user experience.
- Build a classifier model to help predict reviews as positive or negative

Data Sources

Data for this project was obtained from Kaggle [repository]

(https://www.kaggle.com/datasets/datafiniti/consumer-reviews-of-amazon-products? resource=download)

The data represents:

Brand: The brand name of the product being reviewed.

Categories: Categories or tags that classify the product (e.g., electronics, home, books).

Keys: Keywords or identifiers associated with the product.

Manufacturer: The company or entity that manufactures the product.

Reviews.date: The date when the review was posted.

Reviews.dateAdded: Additional date-related information, possibly indicating when the review was added to the dataset.

Reviews.dateSeen: Dates indicating when the review was observed or recorded (possibly by a data aggregator or platform).

Reviews.didPurchase: Boolean (true/false) indicating whether the reviewer claims to have purchased the product.

Reviews.doRecommend: Boolean (true/false) indicating whether the reviewer recommends the product.

Reviews.id: Unique identifier for each review.

Reviews.numHelpful: Number of users who found the review helpful.

Reviews.rating: Rating given by the reviewer (typically on a scale such as 1 to 5 stars).

Reviews.sourceURLs: URLs pointing to the source of the review.

Reviews.text: The main body of the review text.

Reviews.title: The title or headline of the review.

Reviews.userCity: City location of the reviewer.

Reviews.userProvince: Province or state location of the reviewer.

Reviews.username: Username or identifier of the reviewer.

These are the variables this analysis will focus on to derive insights.

Methodology

The process can be divided into these many parts.(we will edit this bit to the exact number once done)

Data preparation

- Text Cleaning: Remove or handle punctuation, special characters, numbers, and stopwords
- Tokenization: Split text into words or subwords.
- Text Normalization: Convert text to lowercase, perform stemming or lemmatization.
- Padding/Truncation: Ensure all text sequences are of the same length.
- Train-Test Split: Divide your data into training, validation, and test sets

EDA Visualisations and insights. For each characteristic we will be:

- Creating visualisations
- Drawing conclusions
- Providing recommendations

Feature Engineering

In the feature engineering section, we process and transform the textual data for further analysis and modeling:

The methods used are;

- Sentiment Analysis
- Visualization with Word Clouds
- Text Vectorization to convert textual data into numerical form using TF-IDF and Count Vectorization.
- Word Embedding using Word2Vec and FastTex

We will also Extract the Bigrams and Trigrams

Model Selection and Building

The models used are a Simple RNN, LSTM, BERT models

Hyperparameter Tuning: Optimize hyperparameters for better performance.

Model Evaluation

Evaluate Performance using the accuracy score.

Analyze Results: Look at the confusion matrix, ROC curves, and other evaluation tools.

Data preparation

Importing Libraries

```
#Basic libraries
import pandas as pd
import numpy as np
#NLTK libraries
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word tokenize
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
import re
import string
!pip install wordcloud
from wordcloud import WordCloud, STOPWORDS
from nltk.stem.porter import PorterStemmer
from sklearn.feature extraction.text import TfidfVectorizer
# Machine Learning libraries
import sklearn
from sklearn.svm import SVC
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.pipeline import make pipeline
from sklearn.model selection import GridSearchCV
from sklearn.linear model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive bayes import BernoulliNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.multiclass import OneVsRestClassifier
from sklearn.svm import SVC
from sklearn.pipeline import Pipeline
from sklearn.model selection import train test split
from sklearn.preprocessing import label binarize
from sklearn import svm, datasets
from sklearn import preprocessing
!pip install tensorflow
!pip install keras
!pip install numpy pandas scikit-learn
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, SimpleRNN, Dense
from tensorflow.keras.preprocessing.text import Tokenizer
```

```
from tensorflow.keras.preprocessing.sequence import pad sequences
#Metrics libraries
from sklearn import metrics
from sklearn.metrics import classification report
from sklearn.model selection import cross val score
from sklearn.metrics import roc auc score
from sklearn.metrics import roc curve, auc
#Visualization libraries
import matplotlib.pyplot as plt
from matplotlib import rcParams
import seaborn as sns
from plotly import tools
import plotly graph objs as go
from plotly.offline import iplot
%matplotlib inline
#Ignore warnings
import warnings
warnings.filterwarnings('ignore')
[nltk data] Downloading package punkt to /root/nltk_data...
              Package punkt is already up-to-date!
[nltk data]
[nltk data] Downloading package stopwords to /root/nltk_data...
              Package stopwords is already up-to-date!
[nltk data]
[nltk data] Downloading package wordnet to /root/nltk data...
              Package wordnet is already up-to-date!
[nltk data]
Requirement already satisfied: wordcloud in
/usr/local/lib/python3.10/dist-packages (1.9.3)
Requirement already satisfied: numpy>=1.6.1 in
/usr/local/lib/python3.10/dist-packages (from wordcloud) (1.25.2)
Requirement already satisfied: pillow in
/usr/local/lib/python3.10/dist-packages (from wordcloud) (9.4.0)
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.10/dist-packages (from wordcloud) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud)
(0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud)
(4.53.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud)
(1.4.5)
```

```
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud)
(24.1)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud)
(3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7-
>matplotlib->wordcloud) (1.16.0)
Requirement already satisfied: tensorflow in
/usr/local/lib/python3.10/dist-packages (2.15.0)
Requirement already satisfied: absl-py>=1.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=23.5.26 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (24.3.25)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1
in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: h5py>=2.9.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.9.0)
Requirement already satisfied: libclang>=13.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: ml-dtypes~=0.2.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: numpy<2.0.0,>=1.23.5 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.25.2)
Requirement already satisfied: opt-einsum>=2.3.2 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.3.0)
Requirement already satisfied: packaging in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (24.1)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!
=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.20.3)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (67.7.2)
Requirement already satisfied: six>=1.12.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.4.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (4.12.2)
Requirement already satisfied: wrapt<1.15,>=1.11.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.14.1)
```

```
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.37.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.64.1)
Requirement already satisfied: tensorboard<2.16,>=2.15 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.15.2)
Requirement already satisfied: tensorflow-estimator<2.16,>=2.15.0
in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.15.0)
Requirement already satisfied: keras<2.16,>=2.15.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.15.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0-
>tensorflow) (0.43.0)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (2.27.0)
Requirement already satisfied: google-auth-oauthlib<2,>=0.5 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (1.2.0)
Requirement already satisfied: markdown>=2.6.8 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (3.6)
Requirement already satisfied: requests<3,>=2.21.0 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (2.31.0)
Reguirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0
in /usr/local/lib/python3.10/dist-packages (from
tensorboard<2.16,>=2.15->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (3.0.3)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.16,>=2.15->tensorflow) (5.3.3)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.16,>=2.15->tensorflow) (0.4.0)
Requirement already satisfied: rsa<5,>=3.1.4 in
/usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.16,>=2.15->tensorflow) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/usr/local/lib/python3.10/dist-packages (from google-auth-
oauthlib<2,>=0.5->tensorboard<2.16,>=2.15->tensorflow) (1.3.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.16,>=2.15->tensorflow) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.16,>=2.15->tensorflow) (3.7)
```

```
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.16,>=2.15->tensorflow) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from reguests<3,>=2.21.0-
>tensorboard<2.16,>=2.15->tensorflow) (2024.6.2)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/usr/local/lib/python3.10/dist-packages (from werkzeug>=1.0.1-
>tensorboard<2.16,>=2.15->tensorflow) (2.1.5)
Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in
/usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1-
>google-auth<3,>=1.6.3->tensorboard<2.16,>=2.15->tensorflow) (0.6.0)
Requirement already satisfied: oauthlib>=3.0.0 in
/usr/local/lib/python3.10/dist-packages (from requests-
oauthlib>=0.7.0->google-auth-oauthlib<2,>=0.5-
>tensorboard<2.16,>=2.15->tensorflow) (3.2.2)
Requirement already satisfied: keras in
/usr/local/lib/python3.10/dist-packages (2.15.0)
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (1.25.2)
Requirement already satisfied: pandas in
/usr/local/lib/python3.10/dist-packages (2.0.3)
Requirement already satisfied: scikit-learn in
/usr/local/lib/python3.10/dist-packages (1.2.2)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2024.1)
Requirement already satisfied: scipy>=1.3.2 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.11.4)
Requirement already satisfied: joblib>=1.1.1 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2-
>pandas) (1.16.0)
```

LOADING DATA

```
# Loading the data set

raw = pd.read_csv('AMAZON REVIEWS.csv')
raw

{"type":"dataframe","variable_name":"raw"}
```

DATA INSPECTION AND UNDERSTANDING

```
# Checking the data types and null values
raw.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23749 entries, 0 to 23748
Data columns (total 21 columns):
#
    Column
                           Non-Null Count Dtype
     -----
 0
    id
                           23749 non-null
                                           obiect
 1
                           23749 non-null
                                           object
    name
 2
                           23747 non-null
                                           object
    asins
 3
    brand
                           23749 non-null
                                           object
 4
                          23749 non-null
                                           object
    categories
 5
    keys
                           23749 non-null
                                           object
 6
                          23749 non-null
    manufacturer
                                           object
 7
                           23722 non-null
    reviews.date
                                           object
 8
    reviews.dateAdded
                          18982 non-null
                                           object
 9
                           23749 non-null
    reviews.dateSeen
                                           object
 10
    reviews.didPurchase
                           1 non-null
                                           object
                           23258 non-null
 11
   reviews.doRecommend
                                           object
   reviews.id
                           1 non-null
 12
                                           float64
                           23291 non-null
 13 reviews.numHelpful
                                           float64
                           23717 non-null
 14 reviews.rating
                                           float64
 15 reviews.sourceURLs
                           23749 non-null
                                           object
 16 reviews.text
                           23748 non-null
                                           object
 17
    reviews.title
                           23746 non-null
                                           object
                          0 non-null
18 reviews.userCitv
                                           float64
 19 reviews.userProvince 0 non-null
                                           float64
20 reviews.username
                           23743 non-null object
dtypes: float64(5), object(16)
memory usage: 3.8+ MB
```

Columns with 0 Non-Null Count

- This column has 0 non-null entries, meaning all 34,660 entries are missing or null.
- This column does not contain any useful data.

Columns with 1 Non-Null Count

- This column has only 1 non-null entry, meaning out of 34,660 rows, only one entry has a value and the rest are null.
- This column contains almost no useful data.

```
# Checking the data shape
raw.shape
(23749, 21)
```

```
#Summary statistics
raw.describe()
{"summary":"{\n \"name\": \"raw\",\n \"rows\": 8,\n \"fields\": [\n
{\n \"column\": \"reviews.id\",\n \"properties\": {\n
\"dtype\": \"number\",\n \"std\": 42094956.36805944,\n
\"min\": 1.0,\n \"max\": 111372787.0,\n
\"num_unique_values\": 2,\n \"samples\": [\n
\"dtype\": \"number\",\n \"std\": 8199.163183167142,\n
\"min\": 0.0,\n \"max\": 23291.0,\n
\"std\":
],\n \"semantic_type\": \"\",\n \"description\": \"\"\n
}\n },\n {\n \"column\":\"reviews.userProvince\",\n
\"properties\": {\n \"dtype\":\"number\",\n \"std\":
null,\n \"min\": 0.0,\n \"max\": 0.0,\n
\"num_unique_values\": 1,\n \"samples\": [\n 0.0\n
],\n \"semantic_type\": \"\",\n \"description\": \"\"\n
}\n }\n ]\n}","type":"dataframe"}
# Previewing the columns
raw.columns
Index(['id', 'name', 'asins', 'brand', 'categories', 'keys',
'manufacturer',
      'reviews.date', 'reviews.dateAdded', 'reviews.dateSeen',
      'reviews.didPurchase', 'reviews.doRecommend', 'reviews.id',
      'reviews.numHelpful', 'reviews.rating', 'reviews.sourceURLs',
      'reviews.text', 'reviews.title', 'reviews.userCity',
      'reviews.userProvince', 'reviews.username'],
     dtype='object')
# Renaming the columns to standard naming convention
column names = {
   'id': 'id',
   'name': 'product name',
```

```
'asins': 'asins',
    'brand': 'brand',
    'categories': 'product_categories',
    'keys': 'product keys',
    'manufacturer': 'manufacturer name',
    'reviews.date': 'review_date',
    'reviews.dateAdded': 'review date added',
    'reviews.dateSeen': 'review date seen',
    'reviews.didPurchase': 'review did purchase',
    'reviews.doRecommend': 'review do recommend',
    'reviews.id': 'review id',
    'reviews.numHelpful': 'review_num_helpful',
    'reviews.rating': 'review rating',
    'reviews.sourceURLs': 'review source urls',
    'reviews.text': 'review_text',
'reviews.title': 'review_title',
    'reviews.userCity': 'review user city',
    'reviews.userProvince': 'review_user_province',
    'reviews.username': 'review username'
}
# Rename columns in your DataFrame
raw.rename(columns=column names, inplace=True)
# Example: Printing the new column names
print(raw.columns)
Index(['id', 'product name', 'asins', 'brand', 'product categories',
       'product_keys', 'manufacturer_name', 'review date',
'review_date_added',
       'review date seen', 'review did purchase',
'review do recommend',
       'review id', 'review num helpful', 'review rating',
       'review source urls', 'review text', 'review title',
'review user city',
       'review user province', 'review username'],
      dtype='object')
# Convert 'review date' to datetime to enable trend analysis
raw['review date'] = pd.to datetime(raw['review date'], format=
'mixed', utc=True)
# Print the data types to verify
raw.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23749 entries, 0 to 23748
Data columns (total 21 columns):
   Column
                            Non-Null Count Dtype
```

```
0
     id
                            23749 non-null
                                            object
 1
                            23749 non-null
                                            object
     product name
 2
     asins
                            23747 non-null
                                            object
 3
                            23749 non-null
     brand
                                            object
 4
     product categories
                            23749 non-null
                                            object
 5
     product keys
                            23749 non-null
                                            object
 6
     manufacturer name
                            23749 non-null
                                            object
 7
                            23722 non-null
     review date
                                            datetime64[ns, UTC]
 8
                            18982 non-null
     review date added
                                            object
 9
     review date seen
                            23749 non-null
                                            object
 10
    review did purchase
                            1 non-null
                                            obiect
 11
     review do recommend
                            23258 non-null
                                            object
     review id
 12
                            1 non-null
                                            float64
 13
    review num helpful
                            23291 non-null
                                            float64
 14 review rating
                            23717 non-null
                                            float64
 15
    review source urls
                            23749 non-null
                                            object
 16
    review text
                            23748 non-null
                                            object
                            23746 non-null
 17
     review_title
                                            object
 18
    review user city
                            0 non-null
                                            float64
     review user province
19
                            0 non-null
                                            float64
20
     review username
                            23743 non-null
                                            object
dtypes: datetime64[ns, UTC](1), float64(5), object(15)
memory usage: 3.8+ MB
# Checking for proportion of missing values
raw.isnull().mean()
id
                         0.000000
product name
                        0.000000
asins
                         0.000084
```

```
brand
                         0.000000
product categories
                         0.000000
product keys
                         0.000000
manufacturer_name
                         0.000000
review date
                         0.001137
review date added
                         0.200724
review date seen
                         0.000000
review did purchase
                         0.999958
review do recommend
                         0.020675
review id
                         0.999958
review num helpful
                         0.019285
review rating
                         0.001347
review source urls
                         0.000000
review text
                         0.000042
review title
                         0.000126
review user city
                         1.000000
review user province
                         1.000000
review username
                         0.000253
dtype: float64
```

```
# Checking the missing values
raw.isnull().sum()
                             0
product name
                             0
                             2
asins
brand
                             0
product categories
                             0
product keys
                             0
manufacturer name
                             0
                            27
review date
review date added
                          4767
review date seen
                             0
review did purchase
                         23748
review do recommend
                           491
review id
                         23748
review num helpful
                           458
                            32
review rating
review source urls
                             0
                             1
review text
review title
                             3
                         23749
review user city
review user province
                         23749
review username
                             6
dtype: int64
#check percentage of missing values
# create a function to check the percentage of missing values
def missing values(raw):
    miss = raw.isnull().sum().sort values(ascending = False)
    percentage miss = (raw.isnull().sum() /
len(raw)).sort values(ascending = False)
    missing = pd.DataFrame({"Missing Values": miss, "Percentage":
percentage miss}).reset index()
    missing.drop(missing[missing["Percentage"] == 0].index, inplace =
True)
    return missing
missing data = missing values(raw)
missing data
{"summary":"{\n \"name\": \"missing data\",\n \"rows\": 13,\n
\"fields\": [\n {\n \"column\": \"index\",\n
\"properties\": {\n \"dtype\": \"string\",\n
\"num_unique_values\": 13,\n \"samples\": [\n
\"asins\",\n \"review_username\",\n
\"review_user_province\"\n ],\n
\"\",\n \"description\": \"\"\n
                                                \"semantic type\":
                                            }\n
                                                     },\n
\"column\": \"Missing Values\",\n \"properties\": {\n
```

```
\"dtype\": \"number\",\n
                               \"std\": 11172,\n
                                                        \"min\": 1,\n
\"max\": 23749,\n
                        \"num unique values\": 11,\n
\"samples\": [\n
                         32,\n
                                        23749,\n
                                                           2\
                     \"semantic type\": \"\",\n
         ],\n
\"description\": \"\"\n
                           {\n
                                                    \"column\":
                     \"properties\": {\n
\"Percentage\",\n
                                                  \"dtype\":
                    \"std\": 0.47043329127704037,\n
\"number\",\n
                                                          \"min\":
4.2107036085729926e-05,\n\\"max\": 1.0,\n
\"num unique values\": 11,\n
                                   \"samples\": [\n
0.0013474251547433576,\n
                                 1.0, n
                                                 8.421407217145985e-
05\n
                      \"semantic type\": \"\",\n
            ],\n
\"description\": \"\"\n
                            }\n }\n ]\
n}","type":"dataframe","variable name":"missing data"}
# Checking for uniques values in all columns
# Loop through each column and print unique values
for column name in raw.columns:
    unique values = raw[column name].unique()
    num unique values = len(unique values)
    print(f"Unique Values in '{column name}' (Total:
{num unique values}):")
    print(unique values)
    print("\n" + "="*50 + "\n")
# change to dataframe
Unique Values in 'id' (Total: 30):
['AVgkIhwDv8e3D10-lebb'
                        'AVqVGZO3nnc1JqDc3jGK' 'AVpe9CMS1cnluZ0-aoC5'
 'AVpfBEWcilAPnD xTGb7'
                        'AVgkIiKWnnc1JgDc3khH'
                                               'AVgkIj9snnc1JgDc3khU'
 'AVsRjfwAU2 QcyX9PHge'
                        'AVqVGZNvQMlqs0JE6eUY'
                                               'AVpfwS CLJeJML43DH5w'
 'AVphgVaX1cnluZ0-DR74'
                        'AVqVGZN9QMlgs0JE6eUZ'
                                               'AVpftoij1cnluZ0-p5n2'
                                               'AVpff7 VilAPnD xc1E
 'AVgkIhxunnc1JgDc3kg '
                        'AVpioXbb1cnluZ0-PImd'
 'AVpjEN4jLJeJML43rpUe' 'AVpg3g4RLJeJML43TxA '
                                              'AVqVGWLKnnc1JqDc3jF1'
 'AV1YnRtnglJLPUi8IJmV' 'AVphPmHuilAPnD x3E5h'
                                               'AVzvXXxbvKc47QAVfRhy'
 'AVpe7AsMilAPnD xQ78G'
                        'AVph0EeEilAPnD x9myq'
                                               'AVakIdntQMlqs0JE6fuB'
 'AVzRlorb-jtxr-f3ygvQ' 'AVqVGWQDv8e3D10-ldFr'
                                              'AVzvXXwEvKc47QAVfRhx'
 'AVpgdkC8ilAPnD xsvyi' 'AV1YnR7wglJLPUi8IJmi' 'AVpfl8cLLJeJML43AE3S']
Unique Values in 'product name' (Total: 33):
['All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi, 16 GB - Includes
Special Offers, Magenta'
 'Kindle Oasis E-reader with Leather Charging Cover - Merlot, 6 High-
Resolution Display (300 ppi), Wi-Fi - Includes Special Offers,,'
 'Amazon Kindle Lighted Leather Cover,,,\r\nAmazon Kindle Lighted
Leather Cover,,,'
 'Amazon Kindle Lighted Leather Cover,,,\r\nKindle Keyboard,,,'
 'Kindle Keyboard,,,\r\nKindle Keyboard,,,'
```

- 'All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi, 32 GB Includes Special Offers, Magenta'
- 'Fire HD 8 Tablet with Alexa, 8 HD Display, 32 GB, Tangerine with Special Offers,'
- 'Amazon 5W USB Official OEM Charger and Power Adapter for Fire Tablets and Kindle eReaders,,,\r\nAmazon 5W USB Official OEM Charger and Power Adapter for Fire Tablets and Kindle eReaders,,,'
- 'All-New Kindle E-reader Black, 6 Glare-Free Touchscreen Display, Wi-Fi Includes Special Offers,,'
- 'Amazon Kindle Fire Hd (3rd Generation) 8gb,,,\r\nAmazon Kindle Fire Hd (3rd Generation) 8gb,,,'
- 'Fire Tablet, 7 Display, Wi-Fi, 8 GB Includes Special Offers, Magenta'
- 'Kindle Oasis E-reader with Leather Charging Cover Black, 6 High-Resolution Display (300 ppi), Wi-Fi Includes Special Offers,,'
- 'Amazon Kindle Voyage 4GB Wi-Fi + 3G Black,,,\r\nAmazon Kindle Voyage 4GB Wi-Fi + 3G Black,,,'
- 'Amazon Kindle Voyage 4GB Wi-Fi + 3G Black,,,\r\nFire HD 8 Tablet with Alexa, 8 HD Display, 16 GB, Tangerine with Special Offers",'
- 'Fire HD 8 Tablet with Alexa, 8 HD Display, 16 GB, Tangerine with Special Offers,'
- 'Amazon Standing Protective Case for Fire HD 6 (4th Generation) Black,,,\r\nAmazon Standing Protective Case for Fire HD 6 (4th Generation) Black,,,'
- 'Certified Refurbished Amazon Fire TV (Previous Generation 1st),,,\r\nCertified Refurbished Amazon Fire TV (Previous Generation 1st),,,'
- 'Brand New Amazon Kindle Fire 16gb 7 Ips Display Tablet Wifi 16 Gb Blue,,,'
- 'Amazon Kindle Touch Leather Case (4th Generation 2011 Release), Olive Green,,,\r\nAmazon Kindle Touch Leather Case (4th Generation 2011 Release), Olive Green,,,'
- 'Fire Kids Edition Tablet, 7 Display, Wi-Fi, 16 GB, Green Kid-Proof Case'
- 'Amazon Kindle Paperwhite eBook reader 4 GB 6 monochrome Paperwhite touchscreen Wi-Fi black,,,'
- 'Kindle Voyage E-reader, 6 High-Resolution Display (300 ppi) with Adaptive Built-in Light, PagePress Sensors, Wi-Fi Includes Special Offers.'
- 'Certified Refurbished Amazon Fire TV Stick (Previous Generation 1st),,,\r\nCertified Refurbished Amazon Fire TV Stick (Previous Generation 1st),,,'
- 'Certified Refurbished Amazon Fire TV Stick (Previous Generation 1st),,,\r\nKindle Paperwhite,,,'
 - 'Kindle Paperwhite,,,\r\nKindle Paperwhite,,,'
- 'Amazon Fire Kids Edition Tablet, 7 Display, Wi-Fi, 16 GB, Blue Kid-Proof Case Blue'
 - 'Kindle Paperwhite E-reader White, 6 High-Resolution Display (300

```
ppi) with Built-in Light, Wi-Fi - Includes Special Offers,,'
 'Amazon Echo and Fire TV Power Adapter,,,\r\nAmazon Echo and Fire TV
Power Adapter,,,'
 'Amazon Fire Hd 8 8in Tablet 16gb Black B018szt3bk 6th Gen (2016)
Android,,,\r\nAmazon Fire Hd 8 8in Tablet 16gb Black B018szt3bk 6th
Gen (2016) Android,,,
 'Certified Refurbished Amazon Fire TV with Alexa Voice Remote,,,\r\
nCertified Refurbished Amazon Fire TV with Alexa Voice Remote,,,
 'Amazon - Fire 16GB (5th Gen, 2015 Release) - Black,,,\r\nAmazon -
Fire 16GB (5th Gen, 2015 Release) - Black,,,
 'Fire Tablet, 7 Display, Wi-Fi, 8 GB - Includes Special Offers,
 'Echo (White),,,\r\nEcho (White),,,']
Unique Values in 'asins' (Total: 30):
['B01AHB9CN2' 'B00VINDBJK' 'B005PB2T0S' 'B002Y27P3M' 'B01AHB9CYG'
 'B01AHB9C1E' 'B01J2G4VBG'
                            'B00ZV9PXP2' 'B0083Q04TA' 'B018Y2290U'
 'B00REQKWGA' 'B00I0YAM4I' 'B018T075DC' nan 'B00DU15MU4' 'B018Y225IA'
 'B005PB2T2Q' 'B018Y23MNM' 'B000QVZDJM' 'B00I0Y8XWQ' 'B00L029KXQ'
 'B00QJDU3KY' 'B018Y22C2Y' 'B01BFIBRIE' 'B01J40RNHU' 'B018SZT3BK'
 'B00UH4D8G2' 'B018Y22BI4' 'B00TSUGXKE' 'B00L9EPT80,B01E6A069U']
Unique Values in 'brand' (Total: 1):
['Amazon']
Unique Values in 'product categories' (Total: 29):
['Electronics, iPad & Tablets, All Tablets, Fire
Tablets, Tablets, Computers & Tablets'
 'eBook Readers, Kindle E-readers, Computers & Tablets, E-Readers &
Accessories, E-Readers'
 'Electronics, eBook Readers & Accessories, Covers, Kindle Store, Amazon
Device Accessories, Kindle E-Reader Accessories, Kindle (5th Generation)
Accessories, Kindle (5th Generation) Covers'
 'Kindle Store, Amazon Devices, Electronics'
 'Tablets, Fire Tablets, Electronics, Computers, Computer Components, Hard
Drives & Storage, Computers & Tablets, All Tablets'
 'Tablets, Fire Tablets, Computers & Tablets, All Tablets'
 'Amazon Devices & Accessories, Amazon Device Accessories, Power
Adapters & Cables, Kindle Store, Kindle E-Reader Accessories, Kindle
Paperwhite Accessories'
 'Electronics, iPad & Tablets, All Tablets, Computers/Tablets &
Networking, Tablets & eBook Readers, Computers & Tablets, E-Readers &
Accessories, E-Readers, Used: Computers
Accessories, Used: Tablets, Computers, iPads Tablets, Kindle E-
```

```
readers, Electronics Features'
 'Computers/Tablets & Networking, Tablets & eBook
Readers, Electronics, eBook Readers & Accessories, eBook Readers'
 'Fire Tablets, Tablets, Computers & Tablets, All Tablets, Electronics,
Tech Toys, Movies, Music, Electronics, iPad & Tablets, Android
Tablets, Frys'
 'Kindle E-readers, Electronics Features, Computers & Tablets, E-Readers
& Accessories, E-Readers, eBook Readers'
 'Computers & Tablets,E-Readers & Accessories,eBook Readers,Kindle E-
readers'
 'Fire Tablets,Tablets,Computers & Tablets,All Tablets'
 'Frys,Software & Books,eReaders & Accessories,Tablet Cases
Covers, Tablet Accessories, Computer Accessories'
 'Electronics,Categories,Streaming Media Players,Amazon Devices'
 'Computers/Tablets & Networking,Tablets & eBook Readers,Computers &
Tablets, Tablets, All Tablets'
 'Amazon Device Accessories,Kindle Store,Kindle Touch (4th Generation)
Accessories, Kindle E-Reader Accessories, Covers, Kindle Touch (4th
Generation) Covers'
 'Walmart for Business,Office
Electronics, Tablets, Office, Electronics, iPad & Tablets, Windows
Tablets, All Windows Tablets, Computers & Tablets, E-Readers &
Accessories, E-Readers, eBook Readers, Kindle E-readers, Computers/Tablets
& Networking, Tablets & eBook Readers, Electronics Features, Books &
Magazines, Book Accessories, eReaders, TVs & Electronics, Computers &
Laptops, Tablets & eReaders'
 'Walmart for Business,Office Electronics,Tablets,Electronics,iPad \&
Tablets, All Tablets, Computers & Tablets, E-Readers & Accessories, Kindle
E-readers, Electronics Features, eBook Readers, See more Amazon Kindle
Voyage (Wi-Fi), See more Amazon Kindle Voyage 4GB, Wi-Fi 3G
(Unlocked...'
 'Electronics, Categories, Fire TV, Kindle Store'
 'mazon.co.uk,Amazon Devices'
 "Electronics, Computers, Computer Accessories, Cases & Bags, Fire
Tablets, Electronics Features, Tablets, Computers & Tablets, Kids'
Tablets, Electronics, Tech Toys, Movies, Music, iPad & Tablets, Top
Rated"
 'Electronics, iPad & Tablets, All Tablets, Computers &
Tablets, Tablets, eBook Readers'
 'Kindle Store,Categories,eBook Readers & Accessories,Fire TV
Accessories, Electronics, Power Adapters & Cables, Amazon Device
Accessories, Power Adapters'
 'Fire Tablets, Tablets, Computers & Tablets, All
Tablets, Computers/Tablets & Networking, Tablets & eBook Readers'
 'Categories,Streaming Media Players,Electronics'
 'Computers & Tablets, Tablets, All Tablets, Computers/Tablets &
Networking, Tablets & eBook Readers, Fire Tablets, Frys'
 'Electronics Features,Fire Tablets,Computers & Tablets,Tablets,All
```

Tablets, Computers/Tablets & Networking, Tablets & eBook Readers'

'Stereos, Remote Controls, Amazon Echo, Audio Docks & Mini Speakers, Amazon Echo Accessories, Kitchen & Dining Features, Speaker Systems, Electronics, TVs Entertainment, Clearance, Smart Hubs & Wireless Routers, Featured Brands, Wireless Speakers, Smart Home & Connected Living, Home Security, Kindle Store, Home Automation, Home, Garage & Office, Home, Voice-Enabled Smart Assistants, Virtual Assistant Speakers, Portable Audio & Headphones, Electronics Features, Amazon Device Accessories, iPod, Audio Player Accessories, Home & Furniture Clearance, Consumer Electronics, Smart Home, Surveillance, Home Improvement, Smart Home & Home Automation Devices, Smart Hubs, Home Safety & Security, Voice Assistants, Alarms & Sensors, Amazon Devices, Audio, Holiday Shop']

Unique Values in 'product_keys' (Total: 30): ['841667104676,amazon/53004484,amazon/b0lahb9cn2,0841667104676,allnewfirehd8tablet8hddisplaywifi16gbincludesspecialoffersmagenta/5620406,allnewfirehd8tablet8hddisplaywifi16gbincludesspecialoffersmagenta/b0lahb9cn2'

- 'kindleoasisereaderwithleatherchargingcovermerlot6highresolutiondispla y300ppiwifiincludesspecialoffers/5234468,amazon/b00vindbjk,kindleoasisereaderwithleatherchargingcovermerlot6highresolu tiondisplay300ppiwifiincludesspecialoffers/b00vindbjk,848719069587,0848719069587'
 - 'amazonkindlelightedleathercover/b005pb2t0s'
 - 'kindlekeyboard/b002y27p3m,amazon/d01101'
- '841667104690,allnewfirehd8tablet8hddisplaywifi32gbincludesspecialoffe rsmagenta/

5620408,0841667104690,allnewfirehd8tablet8hddisplaywifi32gbincludesspecialoffersmagenta/b01ahb9cyg,amazon/53004761'

- 'amazon/b01ahb9c1e,0841667104577,firehd8tabletwithalexa8hddisplay32gbt angerinewithspecialoffers/b01ahb9c1e,firehd8tabletwithalexa8hddisplay32gbtangerinewithspecialoffers/5620411,841667104577'
- '0841667120171,841667120171,amazon5wusbofficialoemchargerpoweradapterforfiretabletskindleereaders/b01j2g4vbg'
- 'allnewkindleereaderblack6glarefreetouchscreendisplaywifiincludesspecialoffers/
- 391843532825, allnewkindleereaderblack6glarefreetouchscreendisplaywifiincludesspecialoffers/
- b00zv9pxp2,0848719083774,allnewkindleereaderblack6glarefreetouchscreen displaywifiincludesspecialoffers/252974470193,amazon/
- b00zv9pxp2,848719083774,allnewkindleereaderblack6glarefreetouchscreend isplaywifiincludesspecialoffers/

322538285013,allnewkindleereaderblack6glarefreetouchscreendisplaywifiincludesspecialoffers/

5442403,allnewkindleereaderblack6glarefreetouchscreendisplaywifiinclud esspecialoffers/

kier2016bk,allnewkindleereaderblack6glarefreetouchscreendisplaywifiinc ludesspecialoffers/

162691587356, allnewkindleereaderblack6glarefreetouchscreendisplaywifiincludesspecialoffers/1631053'

- 'amazon/53000386,amazonkindlefirehd3rdgeneration8gb/122605594245,amazonkindlefirehd3rdgeneration8gb/
- 152615237936,amazonkindlefirehd3rdgeneration8gb/
- 391871762463, amazonkindlefirehd3rdgeneration8gb/b0083q04ta'
- 'firetablet7displaywifi8gbincludesspecialoffersmagenta/5025800,841667103105,0841667103105,amazon/
- b018y229ou,firetablet7displaywifi8gbincludesspecialoffersmagenta/ b018y229ou'
- '0848719057331,kindleoasisereaderwithleatherchargingcoverblack6highres olutiondisplay300ppiwifiincludesspecialoffers/b00reqkwga,amazon/b00reqkwga,kindleoasisereaderwithleatherchargingcoverblack6highresolutiondisplay300ppiwifiincludesspecialoffers/5195001,848719057331'
- 'amazonkindlevoyage4gbwifi3gblack/9301112,amazon/b00ioyam4i,0848719040 098,848719040098,amazonkindlevoyage4gbwifi3gblack/b00ioyam4i'
- 'amazon/b018t075dc,firehd8tabletwithalexa8hddisplay16gbtangerinewithsp ecialoffers/
- 5620410, firehd8tabletwithalexa8hddisplay16gbtangerinewithspecialoffers/b018t075dc,841667103068,0841667103068'
- '848719047530,amazonstandingprotectivecaseforfirehd64thgenerationblack/3610684,amazonstandingprotectivecaseforfirehd64thgenerationblack/018w006857385001p,amazon/
- b00kqe2qaw,amazonstandingprotectivecaseforfirehd64thgenerationblack/018w006857385001'
- '848719035551,0848719035551,certifiedrefurbishedamazonfiretvpreviousge neration1st/b00du15mu4'
- '841667103143,0841667103143,brandnewamazonkindlefire16gb7ipsdisplaytabletwifi16gbblue/
- 5025500, brandnewamazonkindlefire16gb7ipsdisplaytabletwifi16gbblue/b018y225ia, brandnewamazonkindlefire16gb7ipsdisplaytabletwifi16gbblue/201625338826, brandnewamazonkindlefire16gb7ipsdisplaytabletwifi16gbblue/362123960192, amazon/b018y225ia'
- 'amazonkindletouchleathercase4thgeneration2011releaseolivegreen/b005pb 2t2g'

- 'firekidseditiontablet7displaywifi16gbgreenkidproofcase/b018y23mnm,841 667103402,0841667103402,firekidseditiontablet7displaywifi16gbgreenkidproofcase/5026300,amazon/b018y23mnm'
- 'amazon/b00oqvzdjm,848719056099,amazonkindlepaperwhiteebookreader4gb6m onochromepaperwhitetouchscreenwifiblack/
- 263087494445,amazonkindlepaperwhiteebookreader4gb6monochromepaperwhite touchscreenwifiblack/
- 9439005,amazonkindlepaperwhiteebookreader4gb6monochromepaperwhitetouch screenwifiblack/
- b00oqvzdjm,0848719056099,amazonkindlepaperwhiteebookreader4gb6monochromepaperwhitetouchscreenwifiblack/00355266000p'
- '848719040104,kindlevoyageereader6highresolutiondisplay300ppiwithadapt ivebuiltinlightpagepresssensorswifiincludesspecialoffers/b00ioy8xwq,0848719040104,kindlevoyageereader6highresolutiondisplay300p piwithadaptivebuiltinlightpagepresssensorswifiincludesspecialoffers/321689278417,kindlevoyageereader6highresolutiondisplay300ppiwithadapti vebuiltinlightpagepresssensorswifiincludesspecialoffers/9302088,amazon/53002680'
- 'certifiedrefurbishedamazonfiretvstickpreviousgeneration1st/b00lo29kxq,0848719052121,848719052121'
- 'kindlepaperwhite/b00qjdu3ky'
- 'amazon/b018y22c2y,841667103389,0841667103389,firekidseditiontablet7displaywifi16gbbluekidproofcase/
- b018y22c2y,amazonfirekidsedition16qb5thgen2015releaseblue/
- 5026000, amazonfirekidsedition7tablet16gbblue/
- 5026000, amazonkidsedition7inch16gbfiretabletblue/kifk716cblu'
- '841667107868,amazon/53004915,amazonkindlepaperwhitewhite/5435104,0841 667107868,kindlepaperwhiteereaderwhite6highresolutiondisplay300ppiwith builtinlightwifiincludesspecialoffers/b01bfibrie'
 - 'amazonechofiretvpoweradapter/b01j4ornhu,0841667120829,841667120829'
- 'amazonfirehd88intablet16gbblackb018szt3bk6thgen2016android/5538501,amazonfirehd88intablet16gbblackb018szt3bk6thgen2016android/
- b018szt3bk,amazonfirehd88intablet16gbblackb018szt3bk6thgen2016android/182378029308,amazonfirehd88intablet16gbblackb018szt3bk6thgen2016android/
- 322430145717,841667103037,0841667103037,amazonfirehd88intablet16gbblac kb018szt3bk6thgen2016android/152627691815,amazon/
- b018szt3bk,amazonfirehd88intablet16gbblackb018szt3bk6thgen2016android/332403091354,amazonfirehd88intablet16gbblackb018szt3bk6thgen2016android/322598029639'
- 'certifiedrefurbishedamazonfiretvwithalexavoiceremote/b00uh4d8g2,08487 19063264,848719063264'

```
'amazonfire16qb5thgen2015releaseblack/272201222631,amazonfire16qb5thge
n2015releaseblack/
b018y22bi4,841667103129,0841667103129,amazonfire16gb5thgen2015releaseb
lack/5023200,amazonfire16gb5thgen2015releaseblack/
332273296844,amazonfire16gb5thgen2015releaseblack/
232443003172,amazon/b018y22bi4'
'amazon/b00tsugxke,0848719062854,firetablet7displaywifi8gbincludesspec
ialoffersblack/
b00tsugxke,848719062854,firetablet7displaywifi8gbincludesspecialoffers
black/4390200, firetablet7displaywifi8qbincludesspecialoffersblack/
322581680105'
echowhite/263039693056,echowhite/152558276095,echowhite/292178880467,
echowhite/222588935706,echowhite/253120140398,echowhite/
322577436254,echowhite/122597356284,echowhite/132263972952,echowhite/
322586415668, echowhite/152626395386, echowhite/272724680159, echowhite/
222587602421,echowhite/122474318097,echowhite/5588528,echowhite/
112567699636, echowhite/272768463386, echowhite/332175902683, echowhite/
311908601694, echowhite/292041139369, echowhite/192239032596, echowhite/
272768869474,0841667112862,echowhite/222507973621,echowhite/
112391858963, echowhite/291992370210, echowhite/b00l9ept8o, echowhite/
112480241614, echowhite/b01e6ao69u, echowhite/322589755316, echowhite/
322574315372,echowhite/253051886606,echowhite/382165760287,echowhite/
222582493180,echowhite/282581384521,echowhite/112479310908,echowhite/
302201691992,echowhite/201761456849,echowhite/amechow2k,echowhite/
132262816901, echowhite/282571823011, echowhite/
322511136772,841667112862,echowhite/232407174148,echowhite/
322441917397, echowhite/amechow, echowhite/332296207643, echowhite/
152610914446,echowhite/222578584785,echowhite/162591117080,echowhite/
162593787621,echowhite/232407374203,echowhite/162595518416,echowhite/
152623638099,amazon/b01e6ao69u']
Unique Values in 'manufacturer name' (Total: 1):
['Amazon']
Unique Values in 'review_date' (Total: 1001):
<DatetimeArray>
['2017-01-13 00:00:00+00:00', '2017-01-12 00:00:00+00:00',
 '2017-01-23 00:00:00+00:00',
                              '2017-01-24 00:00:00+00:00'
 '2017-01-27 00:00:00+00:00', '2017-02-03 00:00:00+00:00',
 '2017-02-06 00:00:00+00:00', '2017-02-05 00:00:00+00:00'
 '2017-03-20 00:00:00+00:00', '2017-03-19 00:00:00+00:00',
 '2015-07-19 00:00:00+00:00', '2015-07-18 00:00:00+00:00',
```

```
'2015-07-15 00:00:00+00:00', '2015-07-20 00:00:00+00:00', '2015-07-01 00:00:00+00:00', '2017-06-11 00:00:00+00:00', '2017-06-28 00:00:00+00:00', '2017-07-05 00:00:00+00:00', '2017-09-17 23:14:35+00:00', '2017-03-12 21:26:29+00:00']
Length: 1001, dtype: datetime64[ns, UTC]
Unique Values in 'review date added' (Total: 1922):
['2017-07-03T23:33:15Z' '2017-07-03T23:28:24Z' '2017-07-
03T23:27:54Z'
 '2017-05-22T20:53:31Z' '2017-05-22T20:54:21Z' '2017-05-22T20:59:21Z']
Unique Values in 'review date seen' (Total: 1728):
['2017-06-07T09:04:00.000Z,2017-04-30T00:45:00.000Z'
 '2017-06-07T09:04:00.000Z,2017-04-30T00:44:00.000Z'
 '2017-06-07T09:04:00.000Z,2017-04-30T00:42:00.000Z'
 '2017-09-28T00:00:00Z,2017-09-08T00:00:00Z,2017-09-12T00:00:00Z,2017-
08-31T00:00:00Z,2017-08-15T00:00:00Z'
 '2017-09-28T00:00:00Z.2017-09-08T00:00:00Z.2017-09-12T00:00:00Z.2017-
08-31T00:00:00Z,2017-08-08T00:00:00Z,2017-08-15T00:00:00Z,2017-07-
26T00:00:00Z,2017-08-01T00:00:00Z
 '2017-09-28T00:00:00Z,2017-09-08T00:00:00Z,2017-09-12T00:00:00Z,2017-
08-31T00:00:00Z,2017-08-08T00:00:00Z,2017-08-15T00:00:00Z']
Unique Values in 'review did purchase' (Total: 2):
[nan True]
______
Unique Values in 'review do recommend' (Total: 3):
[True False nan]
Unique Values in 'review id' (Total: 2):
            nan 1.11372787e+081
Unique Values in 'review_num_helpful' (Total: 68):
                  3. 55.
                              4. 24. 11.
                                            42.
                                                  62.
[ 0.
       1.
            2.
                                                         7.
                                                              8. 6. 10.
                        5. 271. 730. 221.
                                             53.
  36.
       16.
            15.
                  13.
                                                   nan
                                                         9. 105.
                                                                   19.
                                                                        25.
  21.
       14.
            20.
                 22.
                       12.
                            96. 102.
                                        34. 17. 73. 109.
                                                             27.
                                                                   39.
                                                                        57.
      40.
           33. 112. 355.
                            60. 263. 37. 28. 103. 26.
                                                             32.
                                                                   43.
                                                                        64.
  18.
  23. 650. 780. 740. 139. 126. 69. 75. 48. 292. 144.
```

```
Unique Values in 'review_rating' (Total: 6):
[5. 4. 2. 1. 3. nan]
Unique Values in 'review source urls' (Total: 6984):
['http://reviews.bestbuy.com/3545/5620406/reviews.htm?
format=embedded&page=200,http://reviews.bestbuy.com/3545/5620406/
reviews.htm?format=embedded&page=166'
 'http://reviews.bestbuy.com/3545/5620406/reviews.htm?
format=embedded&page=200,http://reviews.bestbuy.com/3545/5620406/
reviews.htm?format=embedded&page=167'
 'http://reviews.bestbuv.com/3545/5620406/reviews.htm?
format=embedded&page=154,http://reviews.bestbuy.com/3545/5620406/
reviews.htm?format=embedded&page=120'
 'http://reviews.bestbuy.com/3545/5588528/reviews.htm?
format=embedded&page=179,http://reviews.bestbuy.com/3545/5588528/
reviews.htm?format=embedded&page=106,http://reviews.bestbuy.com/
3545/5588528/reviews.htm?format=embedded&page=111,http://
reviews.bestbuy.com/3545/5588528/reviews.htm?
format=embedded&page=80,http://reviews.bestbuy.com/3545/5588528/
reviews.htm?format=embedded&page=12,http://reviews.bestbuy.com/
3545/5588528/reviews.htm?format=embedded&page=19,http://
reviews.bestbuy.com/3545/5588528/reviews.htm?format=embedded&page=4'
 'http://reviews.bestbuy.com/3545/5588528/reviews.htm?
format=embedded&page=179,http://reviews.bestbuy.com/3545/5588528/
reviews.htm?format=embedded&page=106,http://reviews.bestbuy.com/
3545/5588528/reviews.htm?format=embedded&page=111,http://
reviews.bestbuy.com/3545/5588528/reviews.htm?
format=embedded&page=80,http://reviews.bestbuy.com/3545/5588528/
reviews.htm?format=embedded&page=12,http://reviews.bestbuy.com/
3545/5588528/reviews.htm?format=embedded&page=19,http://
reviews.bestbuy.com/3545/5588528/reviews.htm?format=embedded&page=5'
 'http://reviews.bestbuy.com/3545/5588528/reviews.htm?
format=embedded&page=179,http://reviews.bestbuy.com/3545/5588528/
reviews.htm?format=embedded&page=106,http://reviews.bestbuy.com/
3545/5588528/reviews.htm?format=embedded&page=111,http://
reviews.bestbuy.com/3545/5588528/reviews.htm?
format=embedded&page=80,http://reviews.bestbuy.com/3545/5588528/
reviews.htm?format=embedded&page=13,http://reviews.bestbuy.com/
3545/5588528/reviews.htm?format=embedded&page=19,http://
reviews.bestbuy.com/3545/5588528/reviews.htm?format=embedded&page=5']
Unique Values in 'review text' (Total: 23749):
```

```
['This product so far has not disappointed. My children love to use it
and I like the ability to monitor control what content they see with
ease.'
 'great for beginner or experienced person. Bought as a gift and she
loves it'
 'Inexpensive tablet for him to use and learn on, step up from the
NABI. He was thrilled with it, learn how to Skype on it already...'
 'Really enjoy the great speaker and music on demand by just asking
Alex. Great buy!'
 'After plugging my Echo in and downloading the Alexa app the rest of
the process was nice and easy. I just added all my music accounts and
my smart home devices and from then on Alexa has been an amazing
device. Anyone looking for a device that can help with home automation
should get an Echo.'
'My husband loves it!! He likes telling Alexa to play music and to
tell jokes'l
Unique Values in 'review_title' (Total: 13517):
['Kindle' 'very fast' 'Beginner tablet for our 9 year old son.' ...
 'It was a gift for my daughter' 'Amazon Echo is amazing!!!'
 'Great new toy']
Unique Values in 'review user city' (Total: 1):
[nan]
Unique Values in 'review user province' (Total: 1):
[nan]
______
Unique Values in 'review_username' (Total: 18865):
['Adapter' 'truman' 'DaveZ' ... 'AmericanChick' 'Lildave56'
'Destiny13']
```

DATA CLEANING

Handling Missing values

#drop all columns with high percentage of missing values and columns not needed

```
raw.drop(columns = ['review_date_added', 'review date seen',
'review_did_purchase' , 'review_user_city',
'review user province', 'review id' , 'product name' ,
'review source urls'], inplace = True)
# drop rows with missing values
raw.dropna(inplace = True)
# Verify that there are no more missing values
print(raw.isnull().sum().sum()) # Should print 0
# Get the shape of the cleaned data
print(raw.shape)
# Display the first few rows of the cleaned data
raw.head(2)
0
(23251, 13)
{"summary":"{\n \"name\": \"raw\",\n \"rows\": 23251,\n \"fields\":
[\n {\n \"column\": \"id\",\n \"properties\": {\n
\"dtype\": \"category\",\n \"num_unique_values\": 19,\n
\"samples\": [\n \"AVqkIhwDv8e3D10-lebb\",\n
\"AVphgVaX1cnluZ0-DR74\",\n\\"AV1YnRtnglJLPUi8IJmV\"\n
],\n \"semantic type\": \"\",\n \"description\": \"\"\n
                     \"column\": \"asins\",\n
}\n
                                                 \"properties\":
      },\n {\n
          \"dtype\": \"category\",\n
{\n
                                         \"num unique values\":
           \"samples\": [\n
                                    \"B01AHB9CN2\",\n
19,\n
\"B018Y2290U\",\n
                     \"B000QVZDJM\"\n
                                               ],\n
\"semantic type\": \"\",\n \"description\": \"\"\n
          {\n \"column\": \"brand\",\n \"properties\": {\
        \"dtype\": \"category\",\n
                                        \"num unique_values\": 1,\n
n
\"samples\": [\n \"Amazon\"\n
                                           1,\n
\"semantic_type\": \"\",\n \"description\": \"\"\n
                                                           }\
    },\n {\n \"column\": \"product categories\",\n
                        \"dtype\": \"category\",\n
\"properties\": {\n
\"num_unique_values\": 18,\n
                             \"samples\": [\n
\"Electronics,iPad & Tablets,All Tablets,Fire
Tablets, Tablets, Computers & Tablets\"\n
\"semantic type\": \"\",\n \"description\": \"\"\n
                                                           }\
                    \"column\": \"product_keys\",\n
n },\n {\n \"column\": \"product_keys\",\n
\"properties\": {\n \"dtype\": \"category\",\n
\"num unique values\": 19,\n \"samples\": [\n
\"841667104676,amazon/53004484,amazon/b01ahb9cn2,0841667104676,allnewf
irehd8tablet8hddisplaywifi16gbincludesspecialoffersmagenta/
5620406,allnewfirehd8tablet8hddisplaywifi16gbincludesspecialoffersmage
nta/b01ahb9cn2\"\n ],\n \"semantic type\": \"\",\n
                                                 \"column\":
\"description\": \"\"\n
                           }\n
                                  },\n
                                         {\n
```

```
\"manufacturer_name\",\n \"properties\": {\n \"dtype\":
\"category\",\n \"num_unique_values\": 1,\n \"samples\":
[\n \"Amazon\"\n ],\n \"semantic_type\": \"\",\
n \"description\": \"\"\n }\n {\n
\"column\": \"review_date\",\n \"properties\": {\n
\ "dtype\": \"date\",\\n\\": \"2014-10-24 00:00:00+00:00\",\\\\\"
         \"max\": \"2017-10-09 00:00:00+00:00\",\n
\"description\": \"\"\n }\n {\n \"column\":
\"review_do_recommend\",\n \"properties\": {\n \"dtype\":
\"category\",\n \"num_unique_values\": 2,\n \"samples\":
[\n false\n ],\n \"semantic_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"review_num_helpful\",\n \"properties\": {\n \"dtype\"\"number\",\n \"std\": 2.330373955444735,\n \"min\":
                                                                 \"dtype\":
0.0,\n \"max\": 109.0,\n \"num_unique_values\": 49,\n \"samples\": [\n 10.0\n ],\n \"semantic_type\": \"\",\n \"description\": \"\"\n }\n }\n {\n
\"column\": \"review_rating\",\n \"properties\": {\n
\"min\": 1.0,\n \"max\": 5.0,\n \"num unique values\":
}\
\"num unique_values\": 23251,\n \"samples\": [\n
\"It's works really well and don't have any issues at all\"\
         ],\n \"semantic_type\": \"\",\n
\"description\": \"\"\n }\n
                                       },\n {\n
                                                          \"column\":
\"review title\",\n \"properties\": {\n
                                                           \"dtype\":
\"string\",\n \"num_unique_values\": 13283,\n
\"samples\": [\n \"Mostly Love It, a few glitches\"\
n ],\n \"semantic_type\": \"\",\n
\"string\",\n \"num_unique_values\": 18472,\n
\"samples\": [\n \"Bomman26\"\n ],\n
\"semantic_type\": \"\",\n \"description\": \"\"\n
                                                                       }\
     }\n ]\n}","type":"dataframe","variable name":"raw"}
```

Checking for duplicates

```
# Checking duplicated rows
num_duplicated = raw.duplicated().sum()
print(f"Number of duplicated rows: {num_duplicated}")
Number of duplicated rows: 0
```

```
# Checking for duplicates using the 'CustomerId' column
raw[raw.duplicated(subset=["asins"])]
{"summary":"{\n \"name\": \"raw[raw\",\n \"rows\": 23232,\n
\"fields\": [\n {\n \"column\": \"id\",\n \"properties\":
        \"dtype\": \"category\",\n
{\n
                                         \"num unique values\":
19,\n \"samples\": [\n \"AVqkIhwDv8e3D10-lebb\",\n \"AVphgVaX1cnluZ0-DR74\",\n \"AV1YnRtnglJLPUi8IJmV\"\n
      \"semantic_type\": \"\",\n \"description\": \"\"\n \,\n \\"column\": \"asins\",\n \"properties\":
],\n
}\n
         \"dtype\": \"category\",\n \"num_unique_values\":
{\n
          \"samples\": [\n \"B01AHB9CN\(\overline{2}\\)",\n
19,\n
\"B018Y2290U\",\n\\"B000QVZDJM\"\n\],\n
\"semantic_type\": \"\",\n \"description\": \"\"\n
    },\n {\n \"column\": \"brand\",\n \"properties\": {\
       \"dtype\": \"category\",\n \"num_unique_values\": 1,\n
\"samples\": [\n \"Amazon\"\n
                                          ],\n
\"semantic_type\": \"\",\n \"description\": \"\"\n
                                                           }\
n },\n {\n \"column\": \"product_categories\",\n \"properties\": {\n \"dtype\": \"category\",\n
\"num unique values\": 18,\n
                             \"samples\": [\n
\"Electronics,iPad & Tablets,All Tablets,Fire
Tablets, Tablets, Computers & Tablets\"\n
\"semantic_type\": \"\",\n \"description\": \"\"\n
                                                           }\
\"num unique values\": 19,\n \"samples\": [\n
\"841667104676,amazon/53004484,amazon/b01ahb9cn2,0841667104676,allnewf
irehd8tablet8hddisplaywifi16gbincludesspecialoffersmagenta/
5620406,allnewfirehd8tablet8hddisplaywifi16qbincludesspecialoffersmage
nta/b01ahb9cn2\"\n ],\n
                                 \"semantic type\": \"\",\n
\"description\": \"\"\n }\n },\n {\n \"column\":
\"manufacturer_name\",\n \"properties\": {\n \"dtype\":
\"category\",\n \"num_unique_values\": 1,\n \"samples\":
           \"Amazon\"\n ],\n \"semantic_type\": \"\",\
        \"description\": \"\"n }\n
                                         },\n {\n
\"column\": \"review_date\",\n
                                 \"properties\": {\n
\"max\": \"2017-10-07 00:00:00+00:00\",\n
\"num_unique_values\": 901,\n \"samples\": [\n
                                                         \"2017-
\"semantic_type\": \"\",\n
            false\n ],\n
[\n
\"description\": \"\"\n }\n },\n {\n \"column\": \"review_num_helpful\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 2.3312992830645416,\n \"min\":
\"samples\": [\n
                        10.0\n
                                    ],\n
                                                \"semantic type\":
```

```
\"description\": \"\"\n
                                                   },\n
                                         }\n
\"column\": \"review_rating\",\n \"properties\": {\n
\"min\": 1.0,\n
                 \mbox{"max}: 5.0,\n
                                             \"num unique values\":
5,\n \"samples\": [\n 4.0\n ],\n \"semantic_type\": \"\",\n \"description\": \"\"\n
                                                               }\
n },\n {\n \"column\": \"review_text\",\n
\"properties\": {\n \"dtype\": \"string\",\n
\"num unique values\": 23232,\n
                                  \"samples\": [\n
\"Excellent tablet at an exceptional price. I highly recommend this
                            \"semantic_type\": \"\",\n
tablet.\"\n
                  ],\n
\"description\": \"\"\n
                                   },\n {\n
                             }\n
                                                  \"column\":
\"review_title\",\n \"properties\": {\n
                                                    \"dtype\":
\"string\",\n \"num_unique_values\": 13279,\n \"samples\": [\n \"Fine - for Now\"\n ],\n \"semantic_type\": \"\",\n \"description\": \"\"\n
                                                               }\
n },\n {\n \"column\": \"review_username\",\n
\"properties\": {\n \"dtype\": \"string\",\n
\"num unique values\": 18463,\n
                                      \"samples\": [\n
```

- The 'id' column has duplicated rows, but we will not remove them as they reflect valid multiple reviews or transactions for the same product.
- We did not set 'asins' or 'id' as indices because multiple entries for the same product (same 'asins') with different or the same 'id' are common in e-commerce datasets, reflecting multiple reviews or transactions for the same product.

Checking for placeholders

```
Column 'review title': Found 1 occurrences of potential placeholder
'Na'
Column 'review username': Found 1 occurrences of potential placeholder
'none'
Column 'review_username': Found 1 occurrences of potential placeholder
'Unknown'
# Checking our column names
raw.columns
Index(['id', 'asins', 'brand', 'product categories', 'product keys',
        'manufacturer_name', 'review_date', 'review_do_recommend',
'review_num_helpful', 'review_rating', 'review_text',
'review title',
        'review username'],
       dtype='object')
#Checking the null values and data types after changes made
raw.info()
<class 'pandas.core.frame.DataFrame'>
Index: 23251 entries, 0 to 23748
Data columns (total 13 columns):
     Column
                             Non-Null Count Dtype
      -----
 0
     id
                             23251 non-null object
 1
                             23251 non-null object
     asins
 2
     brand
                             23251 non-null object
 3
     product_categories 23251 non-null object
     product_keys 23251 non-null object
manufacturer_name 23251 non-null object
review_date 23251 non-null datetim
 4
 5
 6
     review date
                             23251 non-null datetime64[ns, UTC]
 7
    review do recommend 23251 non-null object
    review_num_helpful 23251 non-null float64
 8
 9
    review rating
                             23251 non-null float64
 10review_text23251 non-null object11review_title23251 non-null object12review_username23251 non-null object
 10 review text
                             23251 non-null object
dtypes: datetime64[ns, UTC](1), float64(2), object(10)
memory usage: 2.5+ MB
```

After cleaning the data set, we now have 34,054 rows and no missing values.

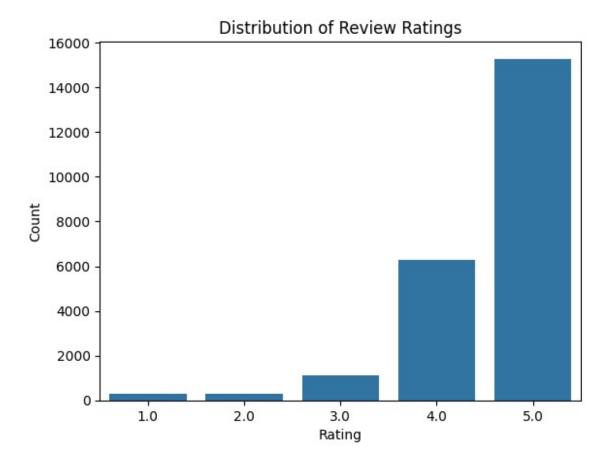
The data set is ready for EDA.

EXPLORATORY DATA ANALYSIS

UNIVARIATE ANALYSIS

1. Distribution of ratings Word frequency, Word cloud and Sentiment Distribution

```
# Distribution of ratings
import matplotlib.pyplot as plt
# Sentiment distribution (simple visualization based on ratings)
sns.countplot(x='review_rating', data=raw)
plt.title('Distribution of Review Ratings')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.show()
```



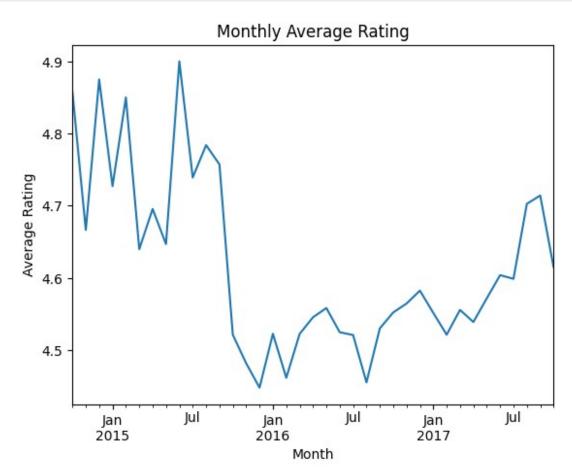
• The distribution of review ratings shows that most reviews tend to be positive, with higher counts towards ratings 4 and 5.

2.Temporal Analysis

```
# Temporal Analysis of rating over time

raw['review_date'] = pd.to_datetime(raw['review_date'])
raw.set_index('review_date', inplace=True)
raw['review_rating'].resample('M').mean().plot()
plt.title('Monthly Average Rating')
```

```
plt.xlabel('Month')
plt.ylabel('Average Rating')
plt.show()
```



• There is a slight fluctuation in average ratings over time, but no clear trend is evident from the monthly average ratings plot.

3. Reviews by product category

```
# Count occurrences of each category
category_counts = raw['product_categories'].value_counts().head(20)

# Extract top 20 categories and their counts
top_categories = category_counts.index

print("Top 20 Product Categories:")
print(category_counts)

# Assuming categories are separated by commas and need to be split
# Convert the 'product_categories' column to string type
raw['product_categories'] = raw['product_categories'].astype(str)
```

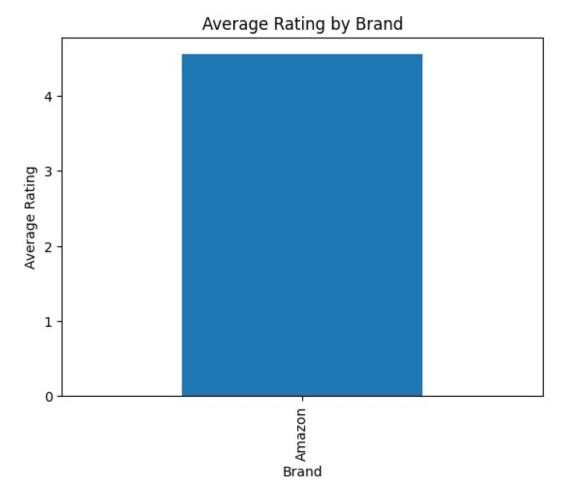
```
# Split the categories by commas
raw['product categories'] = raw['product categories'].str.split(',')
# Explode the list of categories
exploded_raw = raw.explode('product categories')
# Group by 'product_categories' and calculate the mean review rating
mean ratings = exploded raw.groupby('product categories')
['review rating'].mean().sort values(ascending=False)
mean ratings
Top 20 Product Categories:
product categories
Fire Tablets, Tablets, Computers & Tablets, All Tablets, Electronics, Tech
Toys, Movies, Music, Electronics, iPad & Tablets, Android Tablets, Frys
10965
Walmart for Business, Office
Electronics, Tablets, Office, Electronics, iPad & Tablets, Windows
Tablets, All Windows Tablets, Computers & Tablets, E-Readers &
Accessories, E-Readers, eBook Readers, Kindle E-readers, Computers/Tablets
& Networking, Tablets & eBook Readers, Electronics Features, Books &
Magazines, Book Accessories, eReaders, TVs & Electronics, Computers &
Laptops, Tablets & eReaders
3175
Electronics, iPad & Tablets, All Tablets, Fire Tablets, Tablets, Computers
& Tablets
2812
Stereos, Remote Controls, Amazon Echo, Audio Docks & Mini Speakers, Amazon
Echo Accessories, Kitchen & Dining Features, Speaker
Systems, Electronics, TVs Entertainment, Clearance, Smart Hubs & Wireless
Routers, Featured Brands, Wireless Speakers, Smart Home & Connected
Living, Home Security, Kindle Store, Home Automation, Home, Garage &
Office, Home, Voice-Enabled Smart Assistants, Virtual Assistant
Speakers, Portable Audio & Headphones, Electronics Features, Amazon
Device Accessories, iPod, Audio Player Accessories, Home & Furniture
Clearance, Consumer Electronics, Smart Home, Surveillance, Home
Improvement, Smart Home & Home Automation Devices, Smart Hubs, Home
Safety & Security, Voice Assistants, Alarms & Sensors, Amazon
Devices, Audio, Holiday Shop
                                1796
Tablets, Fire Tablets, Computers & Tablets, All Tablets
1698
Computers/Tablets & Networking, Tablets & eBook Readers, Computers &
Tablets, Tablets, All Tablets
1038
Walmart for Business, Office Electronics, Tablets, Electronics, iPad &
Tablets.All Tablets.Computers & Tablets.E-Readers & Accessories.Kindle
E-readers, Electronics Features, eBook Readers, See more Amazon Kindle
Voyage (Wi-Fi), See more Amazon Kindle Voyage 4GB, Wi-Fi 3G
```

```
(Unlocked...
580
Electronics Features, Fire Tablets, Computers & Tablets, Tablets, All
Tablets, Computers/Tablets & Networking, Tablets & eBook Readers
371
Fire Tablets, Tablets, Computers & Tablets, All Tablets, Computers/Tablets
& Networking, Tablets & eBook Readers
269
Electronics,iPad & Tablets,All Tablets,Computers/Tablets &
Networking, Tablets & eBook Readers, Computers & Tablets, E-Readers &
Accessories, E-Readers, Used: Computers
Accessories, Used: Tablets, Computers, iPads Tablets, Kindle E-
readers, Electronics Features
Tablets, Fire Tablets, Electronics, Computers, Computer Components, Hard
Drives & Storage, Computers & Tablets, All Tablets
eBook Readers, Kindle E-readers, Computers & Tablets, E-Readers &
Accessories, E-Readers
67
Computers & Tablets, E-Readers & Accessories, eBook Readers, Kindle E-
readers
51
Electronics, iPad & Tablets, All Tablets, Computers &
Tablets, Tablets, eBook Readers
30
Computers & Tablets, Tablets, All Tablets, Computers/Tablets &
Networking, Tablets & eBook Readers, Fire Tablets, Frys
Fire Tablets, Tablets, Computers & Tablets, All Tablets
Kindle E-readers, Electronics Features, Computers & Tablets, E-Readers &
Accessories, E-Readers, eBook Readers
Electronics, Computers, Computer Accessories, Cases & Bags, Fire
Tablets, Electronics Features, Tablets, Computers & Tablets, Kids'
Tablets, Electronics, Tech Toys, Movies, Music, iPad & Tablets, Top Rated
Name: count, dtype: int64
product categories
Kids' Tablets
                         4.833333
Computer Accessories
                         4.833333
Cases & Bags
                         4.833333
Top Rated
                         4.833333
Tablets & eReaders
                         4.772283
Frys
                         4.454396
Tech Toys
                         4.454380
Music
                         4.454380
```

```
Movies
                        4.454380
Android Tablets
                        4.454172
Name: review_rating, Length: 79, dtype: float64
plt.figure(figsize=(20, 18))
# Create a bar plot with a color gradient
bars = sns.barplot(y=top categories, x=category counts.values,
palette="viridis")
# Add value labels to the bars
for bar, count in zip(bars.patches, category counts.values):
   plt.text(count + 10, # x-coordinate position
             bar.get y() + bar.get height() / 2, # y-coordinate
position
             f'{count}', # formatted label text
             ha='center', va='center', # horizontal and vertical
alignment
             fontsize=10, color='black') # text properties
plt.title('Top 20 Product Categories by Count of Reviews',
fontsize=16)
plt.xlabel('Count', fontsize=14)
plt.ylabel('Product Category', fontsize=14)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.tight layout()
plt.show()
```



```
# Plot review rating by brand
raw.groupby('brand')
['review_rating'].mean().sort_values(ascending=False).plot(kind='bar')
plt.title('Average Rating by Brand')
plt.xlabel('Brand')
plt.ylabel('Average Rating')
plt.show()
```



```
# Assuming categories are separated by commas and need to be split
# Convert the 'product categories' column to string type
raw['product categories'] = raw['product categories'].astype(str)
# Split the categories by commas
raw['product categories'] = raw['product categories'].str.split(',')
# Explode the list of categories
exploded raw = raw.explode('product categories')
# Group by 'product categories' and calculate the mean review rating
mean_ratings = exploded_raw.groupby('product_categories')
['review rating'].mean().sort values(ascending=False)
mean ratings
product categories
 'Kindle E-readers']
                           4.862745
['Computers & Tablets'
                           4.836066
 "Kids' Tablets"
                           4.833333
 'Cases & Bags'
                           4.833333
```

```
'Computer Accessories' 4.833333
...
' Movies' 4.454380
' Tech Toys' 4.454380
' Music' 4.454380
'Android Tablets' 4.454172
['Electronics Features' 4.425876
Name: review_rating, Length: 94, dtype: float64
```

Conclusions

- Fire Tablets, Tablets, Computers & Tablets: Dominates with 10,965 reviews, indicating a strong presence in consumer feedback.
- Stereos, Remote Controls, Amazon Echo: Follows with 6,606 reviews, highlighting significant interest in home electronics and smart devices.
- Back To College, College Electronics: Shows strong engagement in electronics geared towards college students, with 5,051 reviews.

4. Most helpful Votes

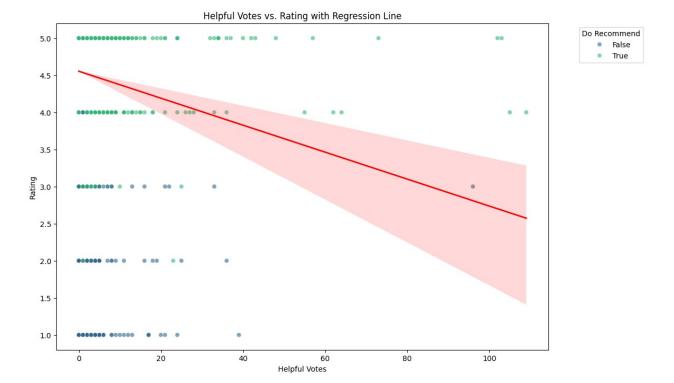
```
# Most helpful reviews
raw.sort values(by='review num helpful', ascending=False).head(10)
{"summary":"{\n \"name\": \"raw\",\n \"rows\": 10,\n \"fields\": [\
    {\n \"column\": \"review_date\",\n \"properties\": {\n
\"dtype\": \"date\",\n \"min\": \"2014-11-16 00:00:00+00:00\",\
        \"max\": \"2016-11-06 00:00:00+00:00\",\n
\"num_unique_values\": 10,\n \"samples\": [\n
                                                          \"2015-
                                 "2016-10-05 00:00:00+00:00",\n
10-01 \ 00:00:00+00:00",\n
\"2015-10-16 00:00:00+00:00\"\n
                                                \"semantic type\":
                                    1,\n
        \"description\": \"\"\n
                                                },\n
                                        }\n
                                                       \{ \n
\"column\": \"id\",\n \"properties\": {\n
                                                   \"dtype\":
\"string\",\n
              \"num unique values\": 5,\n
                                                   \"samples\":
            \"AVqkIiKWnnc1JgDc3khH\",\n
                                               \"AVqkIhwDv8e3D10-
[\n
lebb\",\n
                  \"AV1YnRtnglJLPUi8IJmV\"\n
\"semantic_type\": \"\",\n \"description\": \"\"\n
    \"properties\": {\
        \"dtype\": \"string\",\n \"num_unique_values\": 5,\n
                      \"B01AHB9CYG\",\n
\"B000QVZDJM\"\n
                                                  \"B01AHB9CN2\",\n
\"samples\": [\n
                      ],\n
                                 \"semantic_type\": \"\",\n
\"description\": \"\"\n
                                                  \"column\":
                          }\n
                                 },\n {\n
\"brand\",\n \"properties\": {\n
                                         \"dtype\": \"category\",\
        \"num unique values\": 1,\n \"samples\": [\n
                             \"semantic_type\": \"\",\n
\"Amazon\"\n
                   ],\n
\"description\": \"\"\n }\n },\n {\n \"
\"product_categories\",\n \"properties\": {\n
\"object\",\n \"semantic_type\": \"\",\n
                                        {\n \"column\":
                                                      \"dtype\":
\"description\": \"\"\n }\n
                                                  \"column\":
                                  },\n
                                         {\n
```

```
\"product keys\",\n
                     \"properties\": {\n \"dtype\":
\"string\",\n \"num unique values\": 5,\n
                                                  \"samples\":
[\n
\"841667104690,allnewfirehd8tablet8hddisplaywifi32gbincludesspecialoff
ersmagenta/
5620408,0841667104690,allnewfirehd8tablet8hddisplaywifi32gbincludesspe
cialoffersmagenta/b01ahb9cyg,amazon/53004761\"\n ],\n
                             \"description\": \"\"\n
\"semantic_type\": \"\",\n
                                                         }\
                 \"column\": \"manufacturer name\",\n
           {\n
\"properties\": {\n \"dtype\": \"category\",\n
\"num unique values\": 1,\n
                              \"samples\": [\n
\"Amazon\"\n
                             \"semantic_type\": \"\",\n
                               },\n {\n \"column\":
\"description\": \"\"\n
                          }\n
\"review_do_recommend\",\n \"properties\": {\n
                                                     \"dtvpe\":
\"category\",\n \"num_unique_values\": 2,\n
                                                    \"samples\":
                       ],\n
                                \"semantic type\": \"\",\n
[\n
           false\n
                                      {\n \"column\":
\"description\": \"\"\n
                         }\n
                                },\n
\"review_num_helpful\",\n \"properties\": {\n
                                                    \"dtype\":
\"number\",\n \"std\": 22.237106126672348,\n
                                                    \"min\":
             \"max\": 109.0,\n \"num_unique_values\": 10,\n
55.0.\n
\"samples\": [\n
                       57.0\n
                                   ],\n
                                             \"semantic type\":
            \"description\": \"\"\n
                                       }\n
                                              },\n
\"column\": \"review_rating\",\n \"properties\": {\n
\"min\": 3.0,\n \"max\": 5.0,\n \"num unique values\":
3,\n \"samples\": [\n
                                  4.0\n
                                              ],\n
\"semantic_type\": \"\",\n
                              \"description\": \"\"\n
    },\n {\n \"column\": \"review_text\",\n
\"properties\": {\n \"dtype\": \"string\",\n
\"num unique values\": 10,\n
                            \"samples\": [\n
solid well made tablet that feels good in the hand. images are crisp &
clean. works well with my hulu acct, hbogo acct, prime video acct too.
the first few secs the video is blurry, then it clears up and plays
fine. if you add a sd card the tablet automatically places apps &
video on the sd card (if app allows) the great value for me is you can
download amazon prime video (must be a current member) onto the sd
card for offline viewing. i bet a lot of parents will love that for
those long car trips. sound is good. BT worked well with my BT enabled
stereo ( i streamed prime music & pandora). don't be put off by the
screen resolution - it looks great. the new bellini OS is a MAJOR
improvement over the old carousel format of days past. the new amazon
app underground also adds to the value for prime members. the lock
screen ads are not intrusive, once you swipe to unlock tablet they go
away and do not reappear until you lock the screen again. if you're a
prime member - this is the best 50 bucks you will spend!\"\
                  \"semantic_type\": \"\",\n
\"description\": \"\"\n
                       }\n
                                               \"column\":
                               },\n {\n
\"review_title\",\n \"properties\": {\n
                                               \"dtype\":
\"string\",\n \"num unique values\": 9,\n
                                                  \"samples\":
```

BIVARIATE ANALYSIS

1. Helpful votes vs rating

```
plt.figure(figsize=(12, 8))
# Scatter plot with color coding, size encoding, and transparency
scatter = sns.scatterplot(
    x='review num helpful',
    y='review rating',
    hue='review do recommend',
    sizes=(20, 200), # Minimum and maximum size of points
    alpha=0.6,
    palette='viridis', # Using a different color palette
    data=raw
)
# Add a regression line
sns.regplot(
    x='review num helpful',
    y='review_rating',
    scatter=False,
    color='red',
    line_kws={"linewidth": 2},
    data=raw
)
plt.title('Helpful Votes vs. Rating with Regression Line')
plt.xlabel('Helpful Votes')
plt.vlabel('Rating')
plt.legend(title='Do Recommend', loc='upper right',
bbox to anchor=(1.2, 1)
plt.show()
```



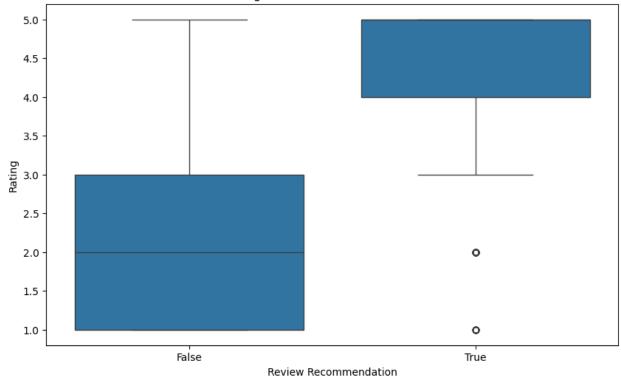
- The scatter plot and regression analysis of helpful votes versus rating illustrate a positive correlation, indicating that more helpful reviews tend to have higher ratings.
- This suggests that customers find high-rated reviews more useful

2. Rating vs. Review recommendation

```
# Convert review_do_recommend to a categorical type
raw['review_do_recommend'] =
raw['review_do_recommend'].astype('category')

# Box plot of rating vs. review recommendation
plt.figure(figsize=(10, 6))
sns.boxplot(x='review_do_recommend', y='review_rating', data=raw)
plt.title('Rating vs. Review Recommendation')
plt.xlabel('Review Recommendation')
plt.ylabel('Rating')
plt.show()
```

Rating vs. Review Recommendation



- The analysis shows that reviews with a positive recommendation (review_do_recommend = True) generally have higher ratings compared to those without a recommendation.
- This highlights the influence of product satisfaction on recommendation.

3. Rating vs Length

```
raw['review_length'] = raw['review_text'].apply(len)
sns.barplot(x='review_rating', y='review_length', data=raw)
plt.title('Review Length vs. Rating')
plt.xlabel('Rating')
plt.ylabel('Review Length')
plt.show()
```

Review Length vs. Rating 250 200 150 50 1.0 2.0 3.0 4.0 5.0

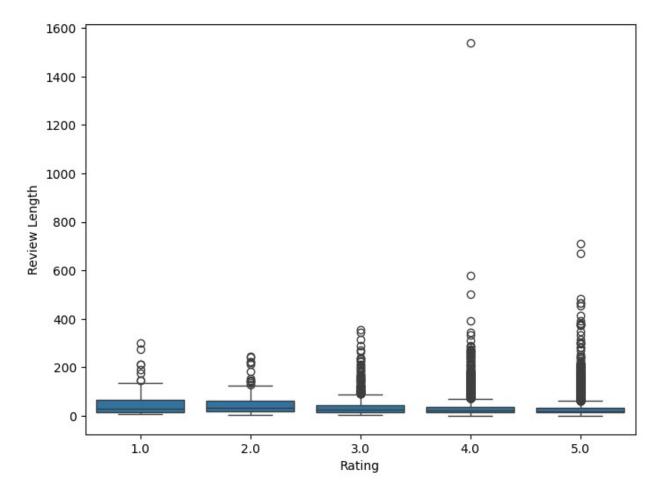
• This visualization illustrates the relationship between review length and review rating. It is evident that shorter reviews tend to receive higher ratings.

Rating

```
word_count=[]
for s1 in raw.review_text:
    word_count.append(len(str(s1).split()))
plt.figure(figsize = (8,6))

import seaborn as sns
import matplotlib.pyplot as plt
sns.boxplot(x="review_rating",y=word_count,data=raw)
plt.xlabel('Rating')
plt.ylabel('Review Length')

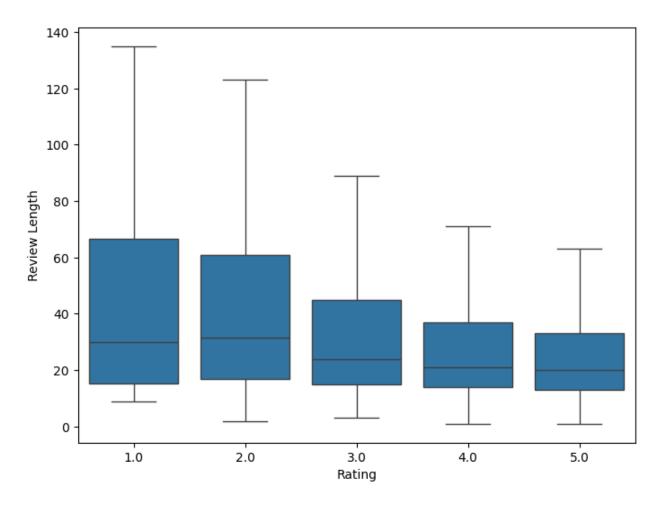
plt.show()
```



• Due to the presence of outliers shown in the box plot, our visualization is currently obscured. To improve clarity, we will proceed by removing these outliers from the dataset.

```
# Generate box plots excluding outliers

plt.figure(figsize = (8,6))
sns.boxplot(x="review_rating",y=word_count,data=raw,showfliers=False)
plt.xlabel('Rating')
plt.ylabel('Review Length')
plt.show()
```



• We can now see that shorter reviews tend to receive higher ratings much better.

Conclusions

The bar plot and box plot analyses show the relationship between review ratings and the length of reviews:

Bar Plot Analysis: Indicates that longer reviews are generally associated with lowerr ratings. This suggests that while longer reviews can provide richer insights, their association with lower ratings indicates that customers who invest more time in detailing their experiences often do so when they feel particularly disappointed or dissatisfied.

Box Plot Analysis: Initially showed outliers affecting clarity in visualization. After excluding outliers, the relationship between review length and rating became clearer

Lower ratings tend to have a wider range of review lengths, suggesting variability in experiences or dissatisfaction reasons.

Higher ratings are associated with a more concentrated range of review lengths, possibly indicating clearer satisfaction or positive experiences with the product.

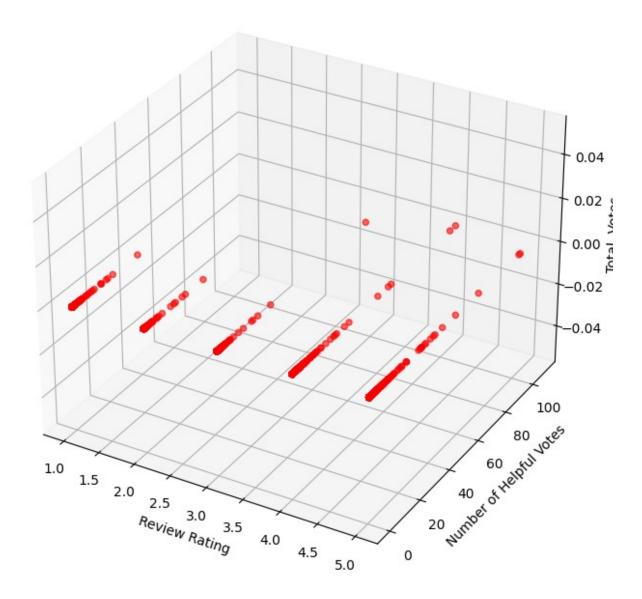
These insights provide a deeper understanding of how review characteristics such as recommendation status and review length correlate with customer ratings, contributing valuable insights for product evaluation and improvement strategies.

3. Multivariate Analysis

1. Scatter plot of reviews

```
# Ensure the column names are correct
review rating col = 'review rating'
review num helpful col = 'review num helpful'
total votes col = 'total votes'
review_did_purchase_col = 'review_did_purchase'
# Check if 'review did purchase' exists, if not create it with a
default value
if review did purchase col not in raw.columns:
    raw[review did purchase col] = False
# Ensure 'total votes' column exists, if not create it with a default
value
if total votes col not in raw.columns:
    raw[total votes col] = 0
# Plotting
fig = plt.figure(figsize=(10, 8))
ax = fig.add subplot(111, projection='3d')
# Map verified purchase to colors
colors = raw[review did purchase col].map({True: 'blue', False:
'red'})
sc = ax.scatter(raw[review rating col], raw[review num helpful col],
raw[total_votes_col], c=colors, alpha=0.6)
# Adding labels and title
ax.set xlabel('Review Rating')
ax.set_ylabel('Number of Helpful Votes')
ax.set zlabel('Total Votes')
plt.title('3D Scatter Plot of Reviews')
plt.show()
```

3D Scatter Plot of Reviews

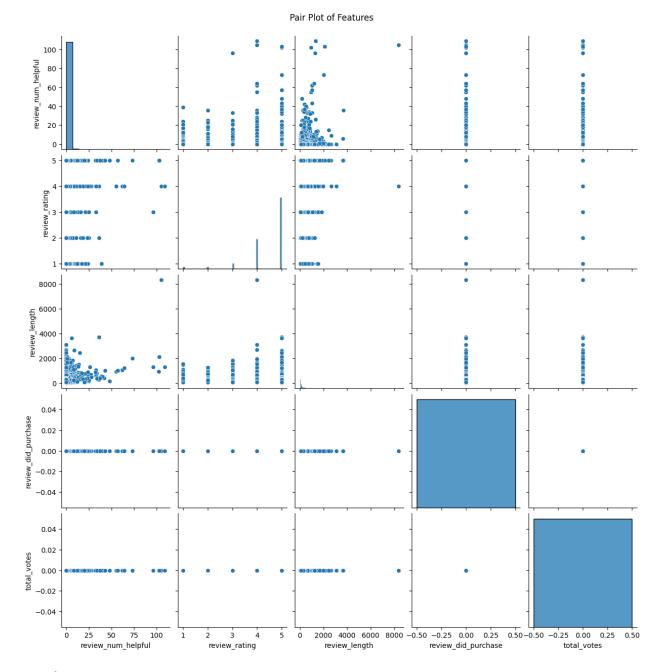


Conclusions

• Visualizing reviews based on rating, helpful votes, and total votes shows various patterns, but it doesn't clearly reveal distinct groups based on whether the purchase was verified.

2. Pair Plot of Features

```
sns.pairplot(raw)
plt.suptitle('Pair Plot of Features', y=1.02)
plt.show()
```



Conclusions

Pair Plot: The pair plot visually explored relationships between different numerical features in the dataset. It provides a quick overview of potential correlations and distributions among variables, aiding in identifying patterns or trends that might warrant further investigation.

Data pre-processing

Check the column names
print(raw.columns)

• Let's preview the first sentence in our text

```
# Previewing the first sentence in our text

first_document = raw.iloc[2]['review_text']
first_document

{"type":"string"}

# Changing the name of our dataframe

data = pd.DataFrame(raw)
```

- For NLP preprocessing, we'll eliminate stopwords, punctuation, and numbers, and convert text to lowercase.
- Subsequently, tokenizing our data is essential because it breaks down text into individual words or tokens, enabling deeper analysis and understanding of the textual content.

```
# Download NLTK stopwords and punctuation
nltk.download('stopwords')
nltk.download('punkt')
# Load stopwords and punctuation
stop words = set(stopwords.words('english'))
# Function to clean and preprocess text
def clean text(text):
    # Ensure text is a string and lowercase
    text = str(text).lower()
    # Remove numbers
    text = re.sub(r'\d+', '', text)
    # Remove punctuation
    text = text.translate(str.maketrans('', '', string.punctuation))
    # Tokenization using regex pattern
    pattern = "([a-zA-Z]+(?:'[a-z]+)?)"
    tokens = nltk.regexp tokenize(text, pattern)
    # Remove stopwords
    clean tokens = [token for token in tokens if token not in
```

```
stop words]
    return ' '.join(clean tokens)
data['clean text'] = raw['review text'].apply(clean text)
data['clean title'] = raw['review title'].apply(clean text)
# Display the cleaned text along with original columns
data[['review text', 'review title', 'clean text', 'clean title']]
[nltk data] Downloading package stopwords to /root/nltk data...
[nltk data]
             Package stopwords is already up-to-date!
[nltk data] Downloading package punkt to /root/nltk data...
[nltk data]
             Package punkt is already up-to-date!
{"summary":"{\n \"name\": \"data[['review text', 'review title',
\"column\": \"review date\",\n \"properties\": {\n
\"dtype\": \"date\",\n \"min\": \"2014-10-24 00:00:00+00:00\",\
        \"max\": \"2017-10-09 00:00:00+00:00\",\n
\"num_unique_values\": 903,\n \"samples\": [\n
                                                            \"2017-
                                 \"2016-04-29 00:00:00+00:00\",\n
03-27 00:00:00+00:00",\n
\"2016-06-10 00:00:00+00:00\"\n
                                     ],\n
                                                \"semantic_type\":
            \"description\": \"\"\n
\"\",\n
                                                 },\n
                                          }\n
                                                        {\n
\"column\": \"review_text\",\n
                                 \"properties\": {\n
\"dtype\": \"string\",\n
                              \"num unique values\": 23251,\n
                        \"It's works really well and don't have any
\"samples\": [\n
issues at all\",\n
                         \"I think this is a great product/
solution, especially for the price. I needed to replace my portable
DVD player. I have 2 young kids and still wanted something to play
movies on the planes and during car trips. After looking at my options
I decided on getting them both a Kindle for about the same price of 1
DVD player. HEADS UP: The Disney Anywhere app takes some work to
download the movies on the kindle but there is a way. You have to
access it through Amazon movies in order to download them. I got a
MicroSD card to store the movies on and so far so good! They held up,
supplied games, movies and when there is Wi-Fi they can play ABCMouse
to get some learning in there too. Great buy. Good luck getting those
movies downloaded :)\",\n
                                 \"This little tablet is marvelous!
I do not have a smart phone and this slips effortlessly in my handbag
to be scanned at a store etc.. It's quick to respond and I can't say
enough about it.\"\n
                           ],\n
                                      \"semantic type\": \"\",\n
\"description\": \"\"\n
                                                  \"column\":
                           }\n
                                  },\n
                                          {\n
                       \"properties\": {\n
\"review title\",\n
                                                  \"dtype\":
                    \"num unique_values\": 13283,\n
\"string\\",\n
                         \"Mostly Love It, a few glitches\",\n
\"samples\": [\n
\"Lots of fun\",\n
                           \"My Wife Loves her E-Reader\"\
                   \"semantic type\": \"\",\n
        ],\n
\"description\": \"\"\n
                           }\n
                                  },\n
                                          {\n
                                                   \"column\":
```

```
\"clean text\",\n
                    \"properties\": {\n
                                             \"dtvpe\":
\"string\",\n
                   \"num unique values\": 23186,\n
\"samples\": [\n
                        \"awesome little tablet kids love use
time\",\n
                 \"bought kindle fire reading kids gaming apps user
friendly quick easy access amazon shopping\",\n \"tablet
child needs tuff content aimed childrens needs keeps entertained long
                        \"semantic type\": \"\",\n
let\"\n
              ],\n
\"description\": \"\"\n
                                 },\n {\n
                                                 \"column\":
                           }\n
\"clean_title\",\n \"properties\": {\n
                                               \"dtype\":
\"category\",\n
                     \"num unique values\": 9359,\n
\"samples\": [\n
                      \"great toy cant tear apart fast sh\",\n
\"great purpose\",\n
\"great purpose\",\n
\"semantic_type\": \"\",\n
                           \"paper white\"\n
                               \"description\": \"\"\n
                                                          }\
    }\n ]\n}","type":"dataframe"}
# Dropping the original columns as we now have the clean ones
data.drop(columns = ['review text', 'review title'] , inplace = True)
data.head(2)
{"summary":"{\n \"name\": \"data\",\n \"rows\": 23251,\n
\"fields\": [\n {\n
                         \"column\": \"review date\",\n
                         \"dtype\": \"date\", \" \" \"":
\"properties\": {\n
\"2014-10-24 00:00:00+00:00\",\n
                                    \"max\": \"2017-10-09
00:00:00+00:00\",\n \"num_unique_values\": 903,\n
\"samples\": [\n
                        \"2017-03-27\ \overline{0}0:00:00+00:00\",\n
\"2016-04-29 00:00:00+00:00\",\n
                                      \"2016-06-10
00:00:00+00:00\"\n ],\n
                                 \"semantic type\": \"\",\n
\"description\": \"\"\n
                       }\n },\n {\n \"column\":
                                       \"dtype\": \"category\",\n
\"id\",\n
             \"properties\": {\n
\"num unique_values\": 19,\n
                                 \"samples\": [\n
\"AVqkIhwDv8e3D10-lebb\",\n
                                 \"AVphgVaX1cnluZ0-DR74\",\n
\"AV1YnRtnglJLPUi8IJmV\"\n
                              ],\n
                                        \"semantic_type\":
\"\",\n \"description\": \"\"\n
                                        }\n
                                               },\n
                                                       {\n
\"column\": \"asins\",\n \"properties\": {\n
                                                     \"dtype\":
\"category\",\n \"num_unique_values\": 19,\n
\"samples\": [\n
\"B000QVZDJM\"\n
                    \"B01AHB9CN2\",\n
                                                 \"B018Y2290U\",\n
                     ],\n \"semantic_type\": \"\",\n
\"description\": \"\"\n
                                },\n {\n \"column\":
                          }\n
\"brand\",\n \"properties\": {\n \"dtype\": \"car
n \"num_unique_values\": 1,\n \"samples\": [\n
                                        \"dtype\": \"category\",\
\"Amazon\"\n
            ],\n
                         \"semantic_type\": \"\",\n
\"description\": \"\"\n
                                        {\n \"column\":
                          }\n
                                },\n
\"product_categories\",\n \"properties\": {\n
                                                   \"dtype\":
\"object\",\n
                  \"semantic_type\": \"\",\n
\"column\":
\"product_keys\",\n \"properties\": {\n
                                                \"dtype\":
\"category\",\n
                     \"num unique values\": 19,\n
\"samples\": [\n
\"841667104676,amazon/53004484,amazon/b01ahb9cn2,0841667104676,allnewf
irehd8tablet8hddisplaywifi16gbincludesspecialoffersmagenta/
```

```
5620406,allnewfirehd8tablet8hddisplaywifi16qbincludesspecialoffersmage
nta/b0lahb9cn2\"\n ],\n \"semantic type\": \"\",\n
\"manufacturer_name\",\n \"properties\": {\n \"dtype\":
\"category\",\n \"num_unique_values\": 1,\n \"samples\":
        \"Amazon\"\n ],\n \"semantic_type\": \"\",\
        \"description\": \"\"\n }\n
                                       },\n
\"column\": \"review_do_recommend\",\n \"properties\": {\n
\"dtype\": \"category\",\n \"num_unique_values\": 2,\n
\"samples\": [\n false\n
                                 ],\n
\"semantic_type\": \"\",\n \"description\": \"\"\n
\"num_unique_values\": 49,\n \"samples\": [\n 10.0\n
],\n \"semantic_type\": \"\",\n \"description\": \"\"\n
\"num_unique_values\": 5,\n \"samples\": [\n
],\n \"semantic_type\": \"\",\n \"description\": \"\"\n
}\n },\n {\n \"column\": \"review_username\",\n
\"properties\": {\n \"dtype\": \"string\",\n
\"num_unique_values\": 18472,\n \"samples\": [\n
\"Bomman26\"\n ],\n
                              \"semantic type\": \"\",\n
\"description\": \"\"\n }\n },\n {\n \"column\":
\"review_length\",\n \"properties\": {\n \"dtype\":
\"number\",\n \"std\": 165,\n \"min\": 6,\n \"max\": 8351,\n \"num_unique_values\": 856,\n \"samples\": [\n 692\n ],\n \"semar
                                             \"semantic type\":
\"\",\n \"description\": \"\"\n }\n },\n {\n
\"column\": \"review_did_purchase\",\n \"properties\": {\n
\"dtype\": \"boolean\",\n\\"num_unique_values\": 1,\n
\"samples\": [\n false\n
                                ],\n
\"semantic_type\": \"\",\n
                             \"description\": \"\"\n
                                                         }\
n },\n {\n \"column\": \"total_votes\",\n
\"properties\": {\n \"dtype\": \"number\",\n
                                                   \"std\":
0,\n \"min\": 0,\n \"max\": 0,\n
\"num_unique_values\": 1,\n \"samples\": [\n
],\n \"semantic type\": \"\",\n \"description\": \"\"\n
\"num_unique_values\": 23186,\n \"samples\": [\n
\"awesome little tablet kids love use time\"\n ],\n
\"semantic_type\": \"\",\n \"description\": \"\"\n
\"num unique values\": 9359,\n \"samples\": [\n
\"great toy cant tear apart fast sh\"\n
                                          ],\n
```

```
\"semantic type\": \"\",\n \"description\": \"\"\n
                                                          }\
    }\n ]\n}","type":"dataframe","variable_name":"data"}
# Rename the columns with the original column names
data.rename(columns={'clean text': 'review text', 'clean title':
'review title'}, inplace=True)
# Display the new DataFrame
data.head(1)
{"summary":"{\n \"name\": \"data\",\n \"rows\": 23251,\n
\"fields\": [\n {\n
                         \"column\": \"review date\",\n
                        \"dtype\": \"date\",\n
\"properties\": {\n
\"2014-10-24 00:00:00+00:00\",\n
                                    \"max\": \"2017-10-09
00:00:00+00:00\",\n \"num_unique_values\": 903,\n
\"samples\": [\n
                        \"2017-03-27 00:00:00+00:00\",\n
\"2016-04-29 00:00:00+00:00\",\n
                                     \"2016-06-10
00:00:00+00:00\"\n ],\n
                                   \"semantic type\": \"\",\n
                                        {\n \"column\":
\"description\": \"\"\n
                           }\n
                                 },\n
\"id\",\n \"properties\": {\n
                                       \"dtype\": \"category\",\n
\"num unique values\": 19,\n
                                 \"samples\": [\n
\"AVqkIhwDv8e3D10-lebb\",\n
                                 \"AVphgVaX1cnluZ0-DR74\",\n
                               ],\n
\"AV1YnRtnglJLPUi8IJmV\"\n
                                        \"semantic type\":
            \"description\": \"\"\n
                                         }\n
                                              },\n {\n
\"column\": \"asins\",\n \"properties\": {\n
                                                     \"dtype\":
\"category\",\n \"num_unique_values\": 19,\n
\"samples\": [\n
\"B000QVZDJM\"\n
                    \"B01AHB9CN2\",\n
                                                 \"B018Y2290U\",\n
                     ],\n
                               \"semantic_type\": \"\",\n
\"description\": \"\"\n
                                },\n {\n \"column\":
                          }\n
\"brand\",\n \"properties\": {\n
                                         \"dtype\": \"category\",\
        \"num_unique_values\": 1,\n
                                        \"samples\": [\n
\"Amazon\"\n
             ],\n \"semantic_type\": \"\",\n
\"description\": \"\"\n
                                        {\n \"column\":
                           }\n
                                 },\n
                                                      \"dtype\":
\"product_categories\",\n \"properties\": {\n
\"object\\",\n\\"semantic type\":\"\",\n
\"description\": \"\"\n }\n
                                                \"column\":
                                },\n {\n
\"product_keys\",\n \"properties\": {\n
\"category\",\n \"num_unique_values\": 19,\n
                                                \"dtype\":
\"samples\": [\n
\"841667104676,amazon/53004484,amazon/b01ahb9cn2,0841667104676,allnewf
irehd8tablet8hddisplaywifi16gbincludesspecialoffersmagenta/
5620406,allnewfirehd8tablet8hddisplaywifi16gbincludesspecialoffersmage
nta/b01ahb9cn2\"\n
                   ],\n
                                \"semantic_type\": \"\",\n
                                         {\n \"column\":
\"description\": \"\"\n }\n },\n {\n
\"manufacturer_name\",\n \"properties\": {\n
\"category\",\n \"num_unique_values\": 1,\n \"samples\"[\n \"Amazon\"\n
                                                   \"samples\":
           \"description\": \"\"\n }\n
                                         },\n
                                                 {\n
\"column\": \"review_do_recommend\",\n
                                         \"properties\": {\n
\"dtype\": \"category\",\n \"num_unique_values\": 2,\n
```

```
\"samples\": [\n false\n
                                                 1,\n
\"semantic_type\": \"\",\n \"description\": \"\"\n }\
n },\n {\n \"column\": \"review_num_helpful\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 2.330373955444735,\n \"min\": 0.0,\n \"max\": 109.0,\n
\"num_unique_values\": 49,\n \"samples\": [\n 10.0\n ],\n \"semantic_type\": \"\",\n \"description\": \"\"\n
],\n \"semantic_type\": \"\",\n \"description\": \"\"\n
}\n },\n {\n \"column\": \"review_username\",\n
\"properties\": {\n \"dtype\": \"string\",\n
\"num_unique_values\": 18472,\n \"samples\": [\n
\"Bomman26\"\n ],\n \"semantic_type\": \"\",\n
\"description\": \"\"\n }\n }\n {\n \"column\": \"review_length\",\n \"properties\": {\n \"dtype\":
\"number\",\n \"std\": 165,\n \"min\": 6,\n \"max\": 8351,\n \"num_unique_values\": 856,\n \"samples\": [\n 692\n ],\n \"semantic_type\":
\"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"review_did_purchase\",\n \"properties\": {\n
\"dtype\": \"boolean\",\n\\"num_unique_values\": 1,\n
\"samples": [\n false\n ],\n
],\n \"semantic_type\": \"\",\n \"description\": \"\"\n
}\n },\n {\n \"column\": \"review_text\",\n
\"properties\": {\n \"dtype\": \"string\",\n
\"num unique values\": 23186,\n \"samples\": [\n
\"awesome little tablet kids love use time\"\n ],\n
\"semantic_type\": \"\",\n \"description\": \"\"\n \\",\n \\"review_title\",\n \\"properties\": \\n \"dtype\": \"category\\",\n \\"num_unique_values\": 9359,\n \\"samples\\": [\n
                                                                             }\
\"great toy cant tear apart fast sh\"\n ],\n
\"semantic_type\": \"\",\n \"description\": \"\"\n
                                                                             }\
      }\n ]\n}","type":"dataframe","variable_name":"data"}
# Download NLTK WordNet
nltk.download('wordnet')
[nltk data] Downloading package wordnet to /root/nltk data...
[nltk_data] Package wordnet is already up-to-date!
True
```

• We will now perform lemmatization, which reduces words to their base form while still preserving their meaning to ensure consistency and improve the accuracy of our analysis.

```
# Initialize the WordNet lemmatizer
lemmatizer = WordNetLemmatizer()
# Initialize the WordNet lemmatizer
lemmatizer = WordNetLemmatizer()
# Function to perform lemmatization on text
def lemmatize text(text):
    words = text.split()
    # Lemmatization
    lemmatized words = [lemmatizer.lemmatize(word) for word in words]
    return ' '.join(lemmatized words)
# Apply lemmatization to review text and review title separately
data['lemmatized_text'] = data['review_text'].apply(lemmatize_text)
data['lemmatized title'] = data['review title'].apply(lemmatize text)
# Display the lemmatized text along with original columns
data[[ 'review text' , 'review title' , 'lemmatized text',
'lemmatized title']]
{"summary":"{\n \"name\": \"data[[ 'review_text' , 'review_title' ,
'lemmatized_text', 'lemmatized_title']]\",\n \"rows\": 23251,\n
\"fields\": [\n
                  {\n
                          \"column\": \"review date\",\n
                          \"dtype\": \"date\", \"
\"properties\": {\n
\"2014-10-24 00:00:00+00:00\",\n
                                       \"max\": \"2017-10-09
00:00:00+00:00\",\n
                         \"num unique_values\": 903,\n
\"samples\": [\n
                         \"2017-03-27 00:00:00+00:00\",\n
\"2016-04-29 00:00:00+00:00\",\n
                                         \"2016-06-10
00:00:00+00:00\"\n
                                    \"semantic type\": \"\",\n
                        ],\n
\"description\": \"\"\n
                            }\n },\n
                                           {\n
                                                 \"column\":
                       \"properties\": {\n
\"review text\",\n
                                                  \"dtype\":
\"string\",\n
                    \"num unique values\": 23186,\n
                         \"awesome little tablet kids love use
\"samples\": [\n
time\",\n
                  \"bought kindle fire reading kids gaming apps user
friendly quick easy access amazon shopping\",\n
                                                        \"tablet
child needs tuff content aimed childrens needs keeps entertained long
let\"\n
              ],\n
                          \"semantic_type\": \"\",\n
\"description\": \"\"\n
                                                    \"column\":
                                   },\n {\n
                            }\n
\"review title\",\n
                       \"properties\": {\n
                                                   \"dtype\":
\"category\",\n
                      \"num unique values\": 9359,\n
\"samples\": [\n
                       \"great toy cant tear apart fast sh\",\n
\"great purpose\",\n
                            \"paper white\"\n
\"semantic type\": \"\",\n \"description\": \"\"\n
                                                              }\
            {\n \"column\": \"lemmatized_text\",\n
    },\n
\"properties\": {\n
                          \"dtype\": \"string\",\n
```

```
\"num unique values\": 23184,\n \"samples\": [\n
\"purchased kindle replace original kindle format dont believe called
paperwhite one without keyboard bottom great kindle however ive always
disappointed back light back light idea managed even reading bed lamp
next bed back light make big difference extra resolution nice really
huge thing youre talking text screen battery life definitely reduced
running back light however point charge every night ill still likely
get several week battery life needing recharge overall youre looking
upgrade kindle get back light huge upgrade help reduce eye strain
important\",\n
                       \"exciting tablet still amazed ease use great
product\",\n
                      \"fire much slogan suggests sensible tablet
inexpensive versatile highly recommend\"\n
\"semantic_type\": \"\",\n \"description\": \"\"\n
                                                                 }\
n },\n {\n \"column\": \"lemmatized_title\",\n \"properties\": {\n \"dtype\": \"category\",\n
\"num unique values\": 9166,\n \"samples\": [\n
\"impressive kid\",\n \"great toddler\",\n
reliable cant beat price\"\n ],\n \"sen
                                                               \"fast
                                                 \"semantic_type\":
               \"description\": \"\"\n }\n
                                                   }\n ]\
n}","type":"dataframe"}
# dropping the columns mot lemmatized
data.drop(columns = ['review text', 'review title'] , inplace = True)
data.head(1)
{"summary":"{\n \"name\": \"data\",\n \"rows\": 23251,\n
                           \"column\": \"review_date\",\n
\"fields\": [\n {\n
                            \"dtype\": \"date\", \"
\"properties\": {\n
\"2014-10-24 00:00:00+00:00\",\n\\\"max\\": \"2017-10-09
00:00:00+00:00\",\n \"num_unique_values\": 903,\n \"samples\": [\n \"2017-03-27 00:00:00+00:00\".\"
                           \"2017-03-27 00:00:00+00:00\",\n
\"samples\": [\n
\"2016-04-29 00:00:00+00:00\",\n
                                          \"2016-06-10
00:00:00+00:00\"\n ],\n
                                     \"semantic type\": \"\",\n
\"description\": \"\"\n
                                            {\n \"column\":
                             }\n
                                     },\n
\"id\",\n
                                           \"dtype\": \"category\",\n
           \"properties\": {\n
\"num_unique_values\": 19,\n
                                     \"samples\": [\n
\"AVqkIhwDv8e3D10-lebb\",\n \"AVph
\"AV1YnRtnglJLPUi8IJmV\"\n ],\n
\"\",\n \"description\": \"\"\n
                                   \"AVphqVaX1cnluZ0-DR74\",\n
                                            \"semantic type\":
                                                    },\n {\n
                                             }\n
\"column\": \"asins\",\n \"properties\": {\n
                                                         \"dtype\":
\"category\",\n \"num_unique_values\": 19,\n \"samples\": [\n \"B01AHB9CN2\",\n \"B000QVZDJM\"\n ],\n \"semantic_type\
                                                       \"B018Y2290U\",\n
                                     \"semantic_type\": \"\",\n
                            n } n }, n {n } "column":
\"description\": \"\"\n
\"brand\",\n \"properties\": {\n \"dtype\": \"cat
n \"num_unique_values\": 1,\n \"samples\": [\n
                                             \"dtype\": \"category\",\
```

```
\"object\",\n
                                         \"semantic type\": \"\",\n
\ensuremath{\mbox{"description}}: \ensuremath{\mbox{"\n}} \ensuremath{\mbox{n}} \ensuremath{\mbox{\mbox{$\backslash$}}}, \ensuremath{\mbox{$\backslash$}} \ensuremath{
                                                                                                            \"column\":
\"product_keys\",\n \"properties\": {\n
                                                                                                             \"dtype\":
\"category\",\n
                                                \"num unique values\": 19,\n
\"samples\": [\n
\"841667104676,amazon/53004484,amazon/b01ahb9cn2,0841667104676,allnewf
irehd8tablet8hddisplaywifi16qbincludesspecialoffersmagenta/
5620406,allnewfirehd8tablet8hddisplaywifi16qbincludesspecialoffersmage
nta/b01ahb9cn2\"\n ],\n
                                                                           \"semantic type\": \"\",\n
\"description\": \"\"\n
                                                                                          {\n \"column\":
                                                      }\n },\n
\"manufacturer_name\",\n \"properties\": {\n \"dtype\":
\"category\",\n \"num_unique_values\": 1,\n \"samples\":
                           [\n
                   \"description\": \"\"\n }\n
\"column\": \"review_do_recommend\",\n \"properties\": {\n
\"dtype\": \"category\",\n \"num unique values\": 2,\n
\"samples\": [\n false\n
                                                                                     ],\n
\"semantic_type\": \"\",\n
                                                                       \"description\": \"\"\n
                                                                                                                                   }\
          },\n {\n \"column\": \"review_num helpful\",\n
\"properties\": {\n
2.330373955444735,\n
                                                      \"dtype\": \"number\\",\n \"std\":
                                                       \"min\": 0.0,\n \"max\": 109.0,\n
\"dtype\": \"number\",\n
\"properties\": {\n \"dtype\": \"number\",\n \"std\": 0.7493758042417185,\n \"min\": 1.0,\n \"max\": 5.0,\n
\"properties\": {\n
\"num_unique_values\": 5,\n \"samples\": [\n
],\n \"semantic_type\": \"\",\n \"description\": \"\"\n
\"dtype\": \"string\",\n
\"properties\": {\n
\"num_unique_values\": 18472,\n \"samples\": [\n
\"Bomman26\"\n ],\n
                                                                       \"semantic_type\": \"\",\n
\"description\": \"\"\n }\n },\n {\n \"column\": \"review_length\",\n \"properties\": {\n \"dtype\":
\"number\",\n \"std\": 165,\n \"min\": 6,\n \"max\": 8351,\n \"num_unique_values\": 856,\n \"samples\": [\n 692\n ],\n \"semar
                                                                                                          \"semantic type\":
\"\",\n \"description\": \"\"\n }\n },\n {\n
\"column\": \"review_did_purchase\",\n \"properties\": {\n
\"dtype\": \"boolean\",\n \"num unique values\": 1,\n
\"samples\": [\n false\n
                                                                                      1,\n
\"semantic type\": \"\",\n \"description\": \"\"\n
                                                                                                                                    }\
\"std\":
0,\n \"min\": 0,\n \"max\": 0,\n
\"num_unique_values\": 1,\n \"samples\": [\n
                                                                                                                                0\n
],\n \"semantic_type\": \"\",\n \"description\": \"\"\n
}\n },\n {\n \"column\": \"lemmatized_text\",\n \"properties\": {\n \"dtype\": \"string\",\n
```

```
\"num unique values\": 23184,\n \"samples\": [\n
\"purchased kindle replace original kindle format dont believe called
paperwhite one without keyboard bottom great kindle however ive always
disappointed back light back light idea managed even reading bed lamp
next bed back light make big difference extra resolution nice really
huge thing youre talking text screen battery life definitely reduced
running back light however point charge every night ill still likely
get several week battery life needing recharge overall youre looking
upgrade kindle get back light huge upgrade help reduce eye strain
important\"\n
                   ],\n
                              \"semantic type\": \"\",\n
\"description\": \"\"\n
                                              {\n \"column\":
                              }\n
                                      },\n
\"lemmatized_title\",\n \"properties\": {\n
                                                           \"dtype\":
\"category\",\n \"num_unique_values\": 9166,\n
\"samples\": [\n \"impressive kid\"\n
\"semantic_type\": \"\",\n \"description\": \"\"\n
                                                                   }\
     }\n ]\n}","type":"dataframe","variable_name":"data"}
# Renaming the lemmatized columns
data.rename(columns={'lemmatized_text': 'review_text',
'lemmatized_title': 'review_title'}, inplace=True)
# Display the new DataFrame
data.head(1)
{"summary":"{\n \"name\": \"data\",\n \"rows\": 23251,\n
\"fields\": [\n {\n
                            \"column\": \"review date\",\n
                            \"dtype\": \"date\",\\n \\"min\":
\"properties\": {\n
                                         \"max\": \"2017-10-09
\"2014-10-24 00:00:00+00:00\",\n
00:00:00+00:00\",\n \"num_unique_values\": 903,\n \"samples\": [\n \"2017-03-27 00:00:00+00:00\",\n
\"2016-04-29 00:00:00+00:00\",\n
                                     \"semantic_type\": \"\",\n
                                            \"2016-06-10
00:00:00+00:00\"\n ],\n
\"id\",\n \"properties\,\"\"samples\": [\n\"num_unique_values\": 19,\n \"AVphgVaX1cnluZ0-DR74\",\n\"\"comantic_type\":
                                            \"dtype\": \"category\",\n
\"AVqkIhwDv8e3D10-lebb\",\n\"AVph\"AV1YnRtnglJLPUi8IJmV\"\n\],\n
                                              \"semantic type\":
\"\",\n \"description\": \"\"\n
                                               }\n
                                                     },\n
                                                              {\n
\"column\": \"asins\",\n \"properties\": {\n
                                                            \"dtype\":
\"category\",\n \"num_unique_values\": 19,\n \"samples\": [\n \"B01AHB9CN2\",\n \"B018Y229\"B000QVZDJM\"\n ],\n \"semantic_type\": \"\",\n
                                                       \"B018Y2290U\",\n
                                     },\n {\n \"column\":
\"description\": \"\"\n }\n
\"brand\",\n \"properties\": {\n \"dtype\": \"category\",\
n \"num_unique_values\": 1,\n \"samples\": [\n
\"Amazon\"\n
              ],\n \"semantic_type\": \"\",\n
                             }\n },\n {\n \"column\":
\"description\": \"\"\n
\"product_categories\",\n \"properties\": {\n \"dtype\":
\"object\",\n \"semantic type\": \"\",\n
\ensuremath{\mbox{"description}}: \ensuremath{\mbox{"\n}} \ensuremath{\mbox{n}} \ensuremath{\mbox{\mbox{$\backslash$}}}, \ensuremath{\mbox{$\backslash$}}
                                                      \"column\":
                                              {\n
```

```
\"product_keys\",\n
                     \"properties\": {\n \"dtype\":
\"category\",\n
                     \"num unique values\": 19,\n
\"samples\": [\n
\"841667104676,amazon/53004484,amazon/b01ahb9cn2,0841667104676,allnewf
irehd8tablet8hddisplaywifi16gbincludesspecialoffersmagenta/
5620406,allnewfirehd8tablet8hddisplaywifi16gbincludesspecialoffersmage
nta/b01ahb9cn2\"\n ],\n
                                 \"semantic type\": \"\",\n
                                        {\n \"column\":
\"description\": \"\"\n }\n },\n {\n
\"manufacturer_name\",\n \"properties\": {\n
\"manutacturer_name\",\n \"properties\": {\n \"dtype\":
\"category\",\n \"num_unique_values\": 1,\n \"samples\":
          \"Amazon\"\n ],\n \"semantic type\": \"\",\
[\n
        \"description\": \"\"\n
                                }\n
                                        },\n
                                                {\n
\"column\": \"review_do_recommend\",\n \"properties\": {\n
\"dtype\": \"category\",\n \"num_unique_values\": 2,\n
\"samples\": [\n false\n
                                 ],\n
\"semantic type\": \"\",\n \"description\": \"\"\n
                                                         }\
\"min\": 0.0,\n \"max\": 109.0,\n
\"num_unique_values\": 49,\n \"samples\": [\n 10.0\n
],\n \"semantic_type\": \"\",\n \"description\": \"\"\n
      },\n {\n \"column\": \"review_rating\",\n
}\n
                        \"dtype\": \"number\",\n
\"properties\": {\n
0.7493758042417185,\n
                                                      \"std\":
                       \"min\": 1.0,\n \"max\": 5.0,\n
\"num unique values\": 5,\n \"samples\": [\n
                                                       4.0\n
],\n \"semantic_type\": \"\",\n \"description\": \"\"\n
}\n },\n {\n \"column\": \"review_username\",\n
\"properties\": {\n \"dtype\": \"string\",\n
\"num_unique_values\": 18472,\n
                               \"samples\": [\n
                               \"semantic_type\": \"\",\n
\"Bomman26\"\n ],\n
\"description\": \"\"\n
                         }\n },\n {\n
                                             \"column\":
                       \"properties\": {\n
\"review_length\",\n
                                                \"dtvpe\":
\"number\",\n \"std\": 165,\n \"min\": 6,\n
                  \"num_unique_values\": 856,\n
\"max\": 8351,\n
                      692\n ],\n
\"samples\": [\n
                                              \"semantic type\":
\"\",\n \"description\": \"\\"n }\n },\n {\n
\"column\\": \"review_did_purchase\\",\n \\"properties\\": {\n
\"dtype\": \"boolean\",\n \"num_unique_values\": 1,\n
\"samples\": [\n false\n
                                 ],\n
\"semantic type\": \"\",\n
                               \"description\": \"\"\n
                                                         }\
\"dtype\": \"number\",\n
                                                    \"std\":
0,\n \"min\": 0,\n \"max\": 0,\n \"num_unique_values\": 1,\n \"samples\": [\n
      \"semantic_type\": \"\",\n \"description\": \"\"\n
],\n
\"num unique values\": 23184,\n \"samples\": [\n
\"purchased kindle replace original kindle format dont believe called
```

```
paperwhite one without keyboard bottom great kindle however ive always
disappointed back light back light idea managed even reading bed lamp
next bed back light make big difference extra resolution nice really
huge thing youre talking text screen battery life definitely reduced
running back light however point charge every night ill still likely
get several week battery life needing recharge overall youre looking
upgrade kindle get back light huge upgrade help reduce eye strain
                                 \"semantic type\": \"\",\n
important\"\n
                     ],\n
                            }\n
\"description\": \"\"\n
                                                       \"column\":
                                     },\n
                                             {\n
\"review_title\",\n \"properties\": {\n \'
\"category\",\n \"num_unique_values\": 9166,\n
\"samples\": [\n \"impressive kid\"\n
                                                      \"dtype\":
\"semantic_type\": \"\",\n \"description\": \"\"\n
                                                                 }\
     }\n ]\n}","type":"dataframe","variable_name":"data"}
# Removing white spaces
# Function to remove extra spaces from text
def remove extra spaces(text):
    return ' '.join(text.strip().split())
# Apply function to the 'lemmatized review text' column
data['clean text'] = data['review text'].apply(remove extra spaces)
# Apply function to the 'lemmatized review title' column
data['clean title'] = data['review title'].apply(remove extra spaces)
# Display cleaned text along with original columns
data[['review text', 'review title','clean text', 'clean title']]
{"summary":"{\n \"name\": \"data[['review text',
'review_title','clean_text', 'clean_title']]\",\n \"rows\": 23251,\n
\"fields\": [\n {\n \"column\": \"review_date\",\n
\"fields\": [\n {\n
\"properties\": {\n
                            \"dtype\": \"date\",\"n
\"2014-10-24 00:00:00+00:00\",\n
                                         \"max\": \"2017-10-09
00:00:00+00:00\",\n
                          \"num unique values\": 903,\n
\"samples\": [\n
                           \"2017-03-27 00:00:00+00:00\",\n
\"2016-04-29 00:00:00+00:00\",\n
                                           \"2016-06-10
00:00:00+00:00\"\n
                                      \"semantic_type\": \"\",\n
                         ],\n
\"description\": \"\"\n
                             }\n },\n {\n
                                                     \"column\":
\"review_text\",\n \"properties\": {\n
                                                     \"dtvpe\":
                     \"num unique_values\": 23184,\n
\"string\",\n
\"samples\": [\n
                          \"purchased kindle replace original kindle
format dont believe called paperwhite one without keyboard bottom
great kindle however ive always disappointed back light back light
idea managed even reading bed lamp next bed back light make big
difference extra resolution nice really huge thing youre talking text
screen battery life definitely reduced running back light however
point charge every night ill still likely get several week battery
life needing recharge overall youre looking upgrade kindle get back
```

```
light huge upgrade help reduce eye strain important\",\n
\"exciting tablet still amazed ease use great product\",\n
\"fire much slogan suggests sensible tablet inexpensive versatile
highly recommend\"\n
                                       \"semantic type\": \"\",\n
                           ],\n
                                                  \"column\":
\"description\": \"\"\n
                           }\n
                                   },\n
                                           {\n
\"review_title\",\n \"properties\": {\n
                                                  \"dtype\":
\"category\",\n
\"samples\": [\n
                      \"num unique values\": 9166,\n
                         \"impressive kid\",\n
                                                       \"great
toddler\",\n
                     \"fast reliable cant beat price\"\n
                                                               ],\n
\"semantic type\": \"\",\n \"description\\": \"\"\n
                                                             }\
            {\n \"column\": \"clean_text\",\n
     },\n
\"properties\": {\n
                          \"dtype\": \"string\",\n
\"num unique values\": 23184,\n
                                      \"samples\": [\n
\"purchased kindle replace original kindle format dont believe called
paperwhite one without keyboard bottom great kindle however ive always
disappointed back light back light idea managed even reading bed lamp
next bed back light make big difference extra resolution nice really
huge thing youre talking text screen battery life definitely reduced
running back light however point charge every night ill still likely
get several week battery life needing recharge overall youre looking
upgrade kindle get back light huge upgrade help reduce eye strain
                       \"exciting tablet still amazed ease use great
important\",\n
product\",\n
                     \"fire much slogan suggests sensible tablet
inexpensive versatile highly recommend\"\n
\"semantic_type\": \"\",\n
                                 \"description\": \"\"\n
                                                             }\
            {\n \"column\": \"clean title\",\n
     },\n
                          \"dtype\": \"category\",\n
\"properties\": {\n
\"num unique values\": 9166,\n
                                     \"samples\": [\n
                              \"great toddler\",\n
\"impressive kid\",\n
                                                           \"fast
reliable cant beat price\"\n
                                               \"semantic type\":
                               ],\n
              \"description\": \"\"\n
\"\",\n
                                          }\n
                                                 }\n 1\
n}","type":"dataframe"}
```

Feature Engineering

In the feature engineering section, we process and transform the textual data for further analysis and modeling:

The methods used are;

- **Sentiment Analysis** to determine the sentiment of each review.
- **Visualization with Word Clouds** to visualize the most frequent words in positive and negative reviews
- **Text Vectorization** to convert textual data into numerical form using TF-IDF and Count Vectorization.
- **Word Embedding** to capture the semantic relationships between words by representing them in a continuous vector space.

Extraction of Bigrams and Trigrams

Sentiment Analysis

This was done using the SentimentIntensityAnalyzer from the vaderSentiment library to calculate a sentiment score for each review.

Each review was labeled with a sentiment score, and reviews were classified as either 'positive' or 'negative' based on this score.

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
# Download the VADER lexicon
nltk.download('vader lexicon')
# Initialize the VADER sentiment analyzer
sid = SentimentIntensityAnalyzer()
# Define the sentiment function to calculate the compound score
def sentiment(x):
    score = sid.polarity scores(x)
    return score['compound']
# Apply the sentiment function to the text column to get sentiment
scores
data['sentiment'] = data['clean text'].apply(lambda x: sentiment(x))
# Print the DataFrame with the sentiment scores
data[['clean text', 'sentiment', 'review rating']]
[nltk data] Downloading package vader lexicon to /root/nltk data...
{"summary":"{\n \"name\": \"data[['clean_text', 'sentiment',
'review rating']]\",\n \"rows\": 23251,\n \"fields\": [\n
\"column\": \"review date\",\n
                                   \"properties\": {\n
\ "dtype\": \"date\",\\n\\": \"2014-10-24 00:00:00+00:00\",\\\\
        \"max\": \"2017-10-09 00:00:00+00:00\",\n
\"num_unique_values\": 903,\n \"samples\": [\n
                                                              \"2017-
03-27 \ 00:00:00+00:00',\n
                                  \"2016-04-29 00:00:00+00:00\",\n
\"2016-06-10 00:00:00+00:00\"\n ],\n
                                                  \"semantic type\":
         \"description\": \"\"\n
                                                  },\n
                                          }\n
                                                          {\n
\"column\": \"clean text\",\n
                                 \"properties\": {\n
\"dtype\": \"string\\",\n
                              \"num unique values\": 23184,\n
                         \"purchased kindle replace original kindle
\"samples\": [\n
format dont believe called paperwhite one without keyboard bottom
great kindle however ive always disappointed back light back light
idea managed even reading bed lamp next bed back light make big
difference extra resolution nice really huge thing youre talking text
screen battery life definitely reduced running back light however
point charge every night ill still likely get several week battery
life needing recharge overall youre looking upgrade kindle get back
light huge upgrade help reduce eye strain important\",\n
```

```
\"exciting tablet still amazed ease use great product\",\n
\"fire much slogan suggests sensible tablet inexpensive versatile
highly recommend\"\n
                           ],\n
                                       \"semantic type\": \"\",\n
\"description\": \"\"\n
                            }\n },\n
                                                     \"column\":
                                            {\n
\"sentiment\",\n \"properties\": {\n \"d\\"number\",\n \"std\": 0.3286495531282306,\n
                                                 \"dtype\":
\"number\",\n\\"std\": 0.3280495551202500,\...\
0.9468,\n\\"max\": 0.9974,\n\\"num_unique_values\":
0.9029
2492,\n
0.3935\n
                                                           0.9029.\n
                       \"semantic_type\": \"\",\n
             ],\n
                           n } n }, n {n } (n ) "column":
\"description\": \"\"\n
\"review_rating\",\n \"properties\": {\n
                                                    \"dtype\":
\"number\\",\n\\"std\": 0.7493758042417185,\n
                                                          \"min\":
             \"max\": 5.0,\n \"num_unique_values\": 5,\n \\n 3.0,\n 2.0\n
1.0, n
\"samples\": [\n
],\n \"semantic_type\": \"\",\n
                                              \"description\": \"\"\n
      }\n ]\n}","type":"dataframe"}
}\n
# Filter the original data DataFrame for negative and positive reviews
negative reviews text = data[data['sentiment'].apply(lambda x: 0 <= x</pre>
<= 0.6)]['clean text']
positive_reviews_text = data[data['sentiment'].apply(lambda x: x >
0.6)]['clean text']
# Create labels for negative and positive reviews
data.loc[data['sentiment'] <= 0.5, 'label'] = 'negative'</pre>
data.loc[data['sentiment'] > 0.5, 'label'] = 'positive'
# Print the updated DataFrame to verify
data[['clean text', 'sentiment', 'label']]
{"summary":"{\n \"name\": \"data[['clean text', 'sentiment',
'label']]\",\n \"rows\": 23251,\n \"fields\": [\n {\n
\"column\": \"review date\",\n
                                    \"properties\": {\n
\"dtype\": \"date\",\n \"min\": \"2014-10-24 00:00:00+00:00\",\
         \"max\": \"2017-10-09 00:00:00+00:00\",\n
\"num_unique_values\": 903,\n \"samples\": [\n
                                 \"2016-04-29 00:00:00+00:00\",\n
03-27 00:00:00+00:00\",\n
\"2016-06-10 00:00:00+00:00\"\n ],\n
                                                   \"semantic type\":
              \"description\": \"\"\n
                                          }\n
                                                  },\n
                                                          {\n
\"column\": \"clean text\",\n
                                  \"properties\": {\n
\"dtype\": \"string\",\n
                           \"num unique values\": 23184,\n
\"samples\": [\n
                          \"purchased kindle replace original kindle
format dont believe called paperwhite one without keyboard bottom
great kindle however ive always disappointed back light back light
idea managed even reading bed lamp next bed back light make big
difference extra resolution nice really huge thing youre talking text
screen battery life definitely reduced running back light however
point charge every night ill still likely get several week battery
```

```
life needing recharge overall youre looking upgrade kindle get back
light huge upgrade help reduce eye strain important\",\n
\"exciting tablet still amazed ease use great product\",\n
\"fire much slogan suggests sensible tablet inexpensive versatile
highly recommend\"\n ],\n \"semantic_type\": \"\",\n
\"description\": \"\"\n
                                                       \"column\":
                                     },\n {\n
                              }\n
\"sentiment\",\n\\"properties\": {\n\\"number\",\n\\"std\": 0.3286495531282306,\n\0.9468,\n\\"max\": 0.9974,\n\\"num_unic
                                                   \"dtype\":
                                                              \"min\": -
               \"max\": 0.9974,\n \"num_unique_values\": \"samples\": [\n 0.123,\n 0.9029
2492,\n
                                                              0.9029, n
                        \"semantic_type\": \"\",\n
0.3935\n
               ],\n
\"description\": \"\"\n }\n
                                   },\n {\n \"column\":
                                              \"dtype\": \"category\",\
\"label\",\n \"properties\": {\n
n \"num_unique_values\": 2,\n
                                              \"samples\": [\n
\"negative\",\n \"positive\"\n ],\n
\"semantic type\": \"\",\n \"description\": \"\"\n
                                                                  }\
     }\n ]\n}","type":"dataframe"}
```

Labelling the reviews using the sentiment scores

- Scores ranging from 0 0.5 will be labeled as **negative**
- Scores ranging from 0.6 1 will be labeled as **positive**

```
# Filter the original data DataFrame for negative and positive reviews
negative reviews text = data[data['sentiment'].apply(lambda x: 0 <= x</pre>
<= 0.5)]['clean text']
positive reviews text = data[data['sentiment'].apply(lambda x: x >
0.5)]['clean text']
# Create labels for negative and positive reviews
data.loc[data['sentiment'] <= 0.5, 'label'] = 'negative'</pre>
data.loc[data['sentiment'] > 0.5, 'label'] = 'positive'
# Print the updated DataFrame to verify
# Print the DataFrame with the sentiment scores
data[['clean_text', 'sentiment', 'label']]
{"summary":"{\n \"name\": \"data[['clean text', 'sentiment',
'label']]\",\n \"rows\": 23251,\n \"fields\": [\n
\"column\": \"review_date\",\n
                                    \"properties\": {\n
\"dtype\": \"date\",\n
                              \"min\": \"2014-10-24 00:00:00+00:00\",\
         \mbox{"max}": \mbox{"2017-10-09 00:00:00+00:00}",\n
\"2016-06-10 00:00:00+00:00\"\n \"2016-04-29 00:00:00+00:00\",\n \"\",\n \"descritti
                                                                \"2017-
                                                    \"semantic type\":
              \"description\": \"\"\n
                                                    },\n
\"column\": \"clean_text\",\n
                                \"properties\": {\n
\"dtype\": \"string\",\n \"num unique values\": 23184,\n
\"samples\": [\n
                          \"purchased kindle replace original kindle
```

```
format dont believe called paperwhite one without keyboard bottom
great kindle however ive always disappointed back light back light
idea managed even reading bed lamp next bed back light make big
difference extra resolution nice really huge thing youre talking text
screen battery life definitely reduced running back light however
point charge every night ill still likely get several week battery
life needing recharge overall youre looking upgrade kindle get back
light huge upgrade help reduce eye strain important\",\n
\"exciting tablet still amazed ease use great product\",\n
\"fire much slogan suggests sensible tablet inexpensive versatile
highly recommend\"\n
                    ],\n
                               \"semantic type\": \"\",\n
\"description\": \"\"\n  }\n
                                },\n {\n
                                              \"column\":
\"sentiment\",\n \"properties\": {\n
                                            \"dtype\":
\"min\": -
                                                   0.9029, n
\"description\":\"\n }\n },\n {\n
                                               \"column\":
\"label\",\n \"properties\": {\n
                                       \"dtype\": \"category\",\
       \"num unique values\": 2,\n
                                      \"samples\": [\n
\"negative\",\n \"positive\"\n ],\n
\"semantic_type\": \"\",\n
                             \"description\": \"\"\n
                                                        }\
    }\n ]\n}","type":"dataframe"}
print("Number of negative reviews:", negative_reviews_text.shape[0])
print("Number of positive reviews:", positive_reviews_text.shape[0])
Number of negative reviews: 4112
Number of positive reviews: 18009
```

• We can observe from this that we have class imbalance.

```
# # DataFrame setup
# data = pd.DataFrame({
# 'clean_text': ["I love this product", "This is the worst thing
ever", "Not bad", "Absolutely fantastic", "Terrible experience"],
# 'sentiment': [0.9, 0.2, 0.6, 0.8, 0.3],
# })

# Create labels for negative and positive reviews
data.loc[data['sentiment'] <= 0.5, 'label'] = 'negative'
data.loc[data['sentiment'] > 0.5, 'label'] = 'positive'

# Filter the original data for negative and positive reviews
negative_reviews_text = data[data['sentiment'].apply(lambda x: 0 <= x
<= 0.5)]['clean_text']
positive_reviews_text = data[data['sentiment'].apply(lambda x: x >
0.5)]['clean_text']
```

```
# Create a CountVectorizer to count word frequencies
vectorizer = CountVectorizer()
# Fit and transform the 'clean text' data for negative and positive
X_negative = vectorizer.fit_transform(negative_reviews_text)
X positive = vectorizer.fit transform(positive reviews text)
# Sum up the counts of each vocabulary word
word frequencies negative = X negative.sum(axis=0).A1
word frequencies positive = X positive.sum(axis=0).A1
# Create a dictionary of word frequencies
vocab = vectorizer.get_feature_names_out()
word frequencies negative = dict(zip(vocab,
word frequencies negative))
word_frequencies_positive = dict(zip(vocab,
word frequencies positive))
# Create word clouds for negative and positive reviews
wordcloud negative = WordCloud(width=800, height=400,
background color='white').generate from frequencies(word frequencies n
egative)
wordcloud positive = WordCloud(width=800, height=400,
background_color='white').generate from frequencies(word frequencies p
ositive)
# Display the word clouds in separate figures
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud negative, interpolation='bilinear')
plt.title('Negative Reviews')
plt.axis('off')
plt.show()
print() # Separating the word clouds display for clarity
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud positive, interpolation='bilinear')
plt.title('Positive Reviews')
plt.axis('off')
plt.show()
# Add a sentiment label column for the countplot
data['sentiment label'] = data['label']
```

Negative Reviews



Positive Reviews



- Let's visualize the distribution of sentiment scores and review ratings.
- We will now convert our labels into numerical data for modeling

```
# Perform label encoding
label encoder = LabelEncoder()
data['labeled'] = label encoder.fit transform(data['label'])
print(data[['clean text', 'sentiment', 'labeled']])
clean text \
review date
2017-01-13 00:00:00+00:00 product far disappointed child love use
like a...
2017-01-13 00:00:00+00:00 great beginner experienced person bought
gift ...
2017-01-13 00:00:00+00:00 inexpensive tablet use learn step nabi
thrille...
2017-01-13 00:00:00+00:00 ive fire hd two week love tablet great
valuewe...
2017-01-12 00:00:00+00:00 bought grand daughter come visit set user
ente...
2017-07-29 00:00:00+00:00 great sound quality great way control smart
de...
2017-07-29 00:00:00+00:00
                                daughter love us every day reminder
question
2017-07-29 00:00:00+00:00
                           really enjoy great speaker music demand
asking...
2017-07-28 00:00:00+00:00
                           plugging echo downloading alexa app rest
proce...
2017-07-28 00:00:00+00:00
                           husband love like telling alexa play music
tel...
                           sentiment labeled
review date
2017-01-13 00:00:00+00:00
                              0.8126
2017-01-13 00:00:00+00:00
                              0.9042
                                            1
2017-01-13 00:00:00+00:00
                              0.4404
                                            0
                                            1
2017-01-13 00:00:00+00:00
                              0.9899
2017-01-12 00:00:00+00:00
                              0.9371
                                            1
```

```
2017-07-29 00:00:00+00:00
                               0.8979
                                             1
2017-07-29 00:00:00+00:00
                               0.6369
                                              1
2017-07-29 00:00:00+00:00
                               0.9144
                                              1
2017-07-28 00:00:00+00:00
                               0.9313
                                              1
2017-07-28 00:00:00+00:00
                               0.8834
                                              1
[23251 rows x 3 columns]
```

Feature Extraction

• In this step, we will extract bigrams from the text data and analyze their frequency.

```
#Extraction of Bigrams
# Function to generate n-grams
from collections import defaultdict
from nltk import ngrams # Import the ngrams function
# Function to generate n-grams
def generate ngrams(clean text, n):
    words = clean text.split()
    return list(ngrams(words, n))
# Initialize a defaultdict for frequency counts
freq dict = defaultdict(int)
# Calculate bigram frequency
for sent in data["clean text"]:
    for word in generate ngrams(sent,2):
        freq dict[word] += 1
# Sort the frequency dictionary and create a DataFrame
fd sorted = pd.DataFrame(sorted(freq dict.items(), key=lambda x: x[1],
reverse=True))
fd sorted.columns = ["word", "wordcount"]
print(fd sorted.head(25))
                    word wordcount
0
             (easy, use)
                                1752
1
             (year, old)
                                1261
2
          (kindle, fire)
                                 873
3
         (great, tablet)
                                 865
4
          (great, price)
                                 674
5
           (work, great)
                                 671
6
         (battery, life)
                                 640
7
            (play, game)
                                 554
8
          (amazon, fire)
                                 493
9
          (fire, tablet)
                                 483
10
                                 483
             (best, buy)
11
         (tablet, great)
                                 481
12
            (read, book)
                                 479
```

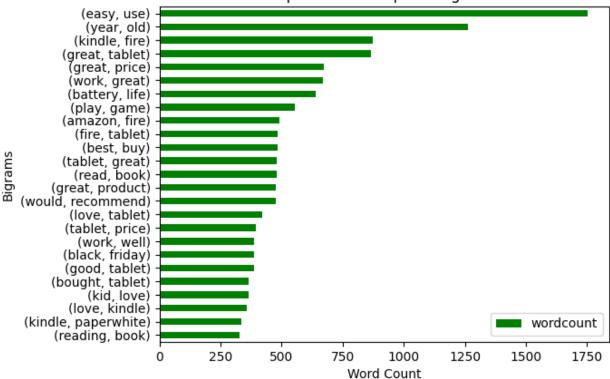
```
13
        (great, product)
                                  478
14
      (would, recommend)
                                  476
15
          (love, tablet)
                                  420
         (tablet, price)
                                  394
16
             (work, well)
17
                                  389
         (black, friday)
                                  388
18
19
          (good, tablet)
                                  386
20
        (bought, tablet)
                                  365
21
             (kid, love)
                                 364
22
          (love, kindle)
                                 357
23
   (kindle, paperwhite)
                                 334
24
         (reading, book)
                                 327
```

• Let's visualize the top 25 most frequent bigrams

```
# Function to plot a horizontal bar chart
def horizontal_bar_chart(data, color):
    data.plot(kind='barh', x='word', y='wordcount', color=color)
    plt.xlabel('Word Count')
    plt.ylabel('Bigrams')
    plt.title('Top 25 Most Frequent Bigrams')
    plt.gca().invert_yaxis() # Invert y-axis to have the highest
count on top
    plt.show()

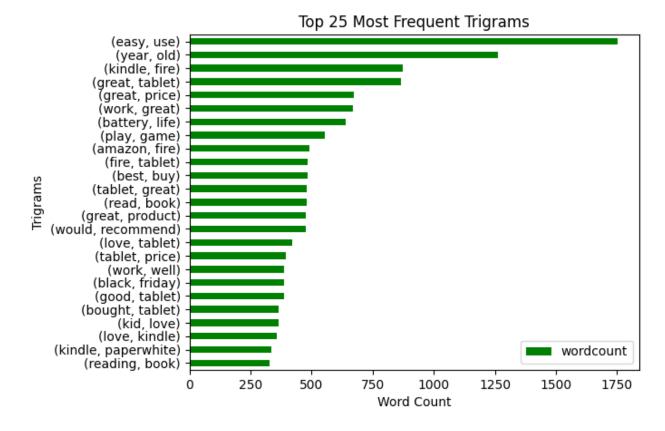
# Plot the top 25 most frequent bigrams
horizontal_bar_chart(fd_sorted.head(25), 'green')
```





```
#Extraction of Trigrams
# Calculate trigram frequency
for sent in data["clean text"]:
    for word in generate ngrams(sent,3):
        freq dict[word] += 1
# Sort the frequency dictionary and create a DataFrame
fd_sorted = pd.DataFrame(sorted(freq_dict.items(), key=lambda x: x[1],
reverse=True))
fd_sorted.columns = ["word", "wordcount"]
print(fd sorted.head(25))
                     word
                           wordcount
0
              (easy, use)
                                 1752
1
              (year, old)
                                 1261
2
          (kindle, fire)
                                  873
3
         (great, tablet)
                                  865
4
          (great, price)
                                  674
5
            (work, great)
                                  671
6
         (battery, life)
                                  640
7
             (play, game)
                                  554
8
          (amazon, fire)
                                  493
9
          (fire, tablet)
                                  483
10
              (best, buy)
                                  483
```

```
11
         (tablet, great)
                                 481
12
            (read, book)
                                 479
13
        (great, product)
                                 478
14
      (would, recommend)
                                 476
15
          (love, tablet)
                                 420
         (tablet, price)
16
                                 394
17
                                389
            (work, well)
         (black, friday)
18
                                 388
          (good, tablet)
19
                                386
20
        (bought, tablet)
                                365
21
             (kid, love)
                                364
22
          (love, kindle)
                                357
23
    (kindle, paperwhite)
                                 334
24
         (reading, book)
                                327
# Function to plot a horizontal bar chart
def horizontal bar chart(data, color):
    data.plot(kind='barh', x='word', y='wordcount', color=color)
    plt.xlabel('Word Count')
    plt.ylabel('Trigrams')
    plt.title('Top 25 Most Frequent Trigrams')
    plt.gca().invert yaxis() # Invert y-axis to have the highest
count on top
    plt.show()
# Plot the top 25 most frequent trigrams
horizontal bar chart(fd sorted.head(25), 'green')
```



Word Vectorization

Methods used are:

TF-IDF Vectorization

The TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer transforms the text into a weighted matrix, where each term's importance is adjusted based on its frequency in the document and across all documents.

Count Vectorization

The Count Vectorizer to converts the text into a matrix of token counts, representing the raw frequency of each term.

The result

Two matrices one with TF-IDF weights and another with raw token counts, each representing the reviews in a numerical format.

```
from sklearn.feature_extraction.text import CountVectorizer

clean_text = data['clean_text']

# Initialize CountVectorizer
vectorizer = CountVectorizer()
```

```
# Fit and transform the clean text column
X count = vectorizer.fit transform(clean text)
# Print the array representation of the features
print(X count.toarray()[1:])
[[0 0 0 ... 0 0 0]
 [0 \ 0 \ 0 \ \dots \ 0 \ 0 \ 0]
 [0 \ 0 \ 0 \ \dots \ 0 \ 0 \ 0]
 [0 \ 0 \ 0 \ \dots \ 0 \ 0 \ 0]
 [0 \ 0 \ 0 \ \dots \ 0 \ 0 \ 0]
 [0 0 0 ... 0 0 0]]
# CountVectorizer
count vec = CountVectorizer()
# Convert the Pandas Series to a list of strings
X_count = count_vec.fit_transform(clean_text.tolist())
print('CountVectorizer:')
print(count vec.get feature names out()[:10], '\n')
CountVectorizer:
['aa' 'abandon' 'abandoned' 'abattery' 'abc' 'abcmouse' 'abcmousecom'
 'abd' 'ability' 'abilty']
```

We extracted the first 10 feature names

Next is the TF-IDF Vectorizer

```
from sklearn.feature_extraction.text import TfidfVectorizer
#Initialize the TfidfVectorizer
vectorizer = TfidfVectorizer()

# Fit the vectorizer to the corpus and transform the corpus into a TF-
IDF matrix
X_tfidf = vectorizer.fit_transform(clean_text)

# Print the TF-IDF matrix as a dense array
print(X_tfidf.toarray(), "\n")

# Print the feature names
print("Feature names:")
print(vectorizer.get_feature_names_out())

[[0. 0. 0. ... 0. 0. 0.]
[0. 0. 0. ... 0. 0. 0.]
[0. 0. 0. ... 0. 0. 0.]
[0. 0. 0. ... 0. 0. 0.]
[0. 0. 0. ... 0. 0. 0.]
[0. 0. 0. ... 0. 0. 0.]
[0. 0. 0. ... 0. 0. 0.]
[0. 0. 0. ... 0. 0. 0.]
[0. 0. 0. ... 0. 0.]
[0. 0. 0. ... 0. 0.]
[0. 0. 0. ... 0. 0.]
[0. 0. 0. ... 0. 0.]
[0. 0. 0. ... 0. 0.]
[0. 0. 0. ... 0. 0.]
[0. 0. 0. ... 0. 0.]
[0. 0. 0. ... 0. 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.]
[0. 0. 0. ... 0.
```

```
[0. 0. 0. ... 0. 0. 0.]
[0. 0. 0. ... 0. 0. 0.]
[0. 0. 0. ... 0. 0. 0.]]

Feature names:
['aa' 'abandon' 'abandoned' ... 'zoomed' 'zooming' 'zwave']
```

Word Embedding Techniques (Word2Vec and FastText):

We used advanced word embedding techniques to capture the semantic meaning of words in the reviews.

Word2Vec: This technique uses a neural network model to learn vector representations of words based on their context in the corpus. We trained a Word2Vec model on our tokenized text data to obtain word vectors.

FastText: Similar to Word2Vec, but it also considers subword information, making it better at handling rare and out-of-vocabulary words. We trained a FastText model to generate word vectors that include subword information.

```
from gensim.models import Word2Vec
from nltk.tokenize import word_tokenize
# Tokenize the text
sentences = [word tokenize(doc.lower()) for doc in data['clean text']]
# Train Word2Vec model
model = Word2Vec(sentences, vector size=100, window=5, min count=1,
workers=4)
# Get word vectors
word vectors = model.wv
# Get the combined matrix of word vectors
wordvec matrix = word_vectors.vectors
print(wordvec matrix)
[[-5.91852427e-01 -1.15684964e-01 7.11060837e-02 ... -4.64988738e-01
   8.14194262e-01 3.35355252e-01]
 [-1.52135766e+00 7.26664364e-01 5.14629662e-01 ... -1.70367733e-01
  -1.11762919e-01 -2.52520949e-01]
 [-8.51729035e-01 \quad 8.20121348e-01 \quad -3.19332480e-01 \quad \dots \quad -7.64260054e-01
   5.89437708e-02 -1.85080305e-01]
 [-2.01675110e-02 -1.57647708e-03 -3.30265216e-03 ... -1.41850607e-02
   8.24625138e-03 -5.22816647e-03]
 [-3.46319191e-03 -5.92788681e-04 5.84608503e-03 ... -2.08582189e-02
 5.52937156e-03 3.03241285e-03]
[-9.75433458e-03 1.20619293e-02 -9.40152165e-03 ... -2.92368070e-03
   9.76519287e-03 4.43352619e-03]]
```

```
from gensim.models import FastText
from nltk.tokenize import word tokenize
# Tokenize the text
sentences = [word tokenize(doc.lower()) for doc in data['clean text']]
# Train FastText model
model = FastText(sentences, vector size=100, window=5, min count=1,
workers=4)
# Get word vectors
word vectors = model.wv
# Get the combined matrix of word vectors
fasttext matrix = word_vectors.vectors
print(fasttext matrix)
[[-0.9189425
              0.48709598 -0.76550424 ... -0.3025607 0.46892732
  -0.0281456 ]
 [-1.5981714 -0.71410924 -0.8901838 ... 0.48710644 1.2479365
   0.5013436 1
 [-1.2803963 -0.25787374 -0.9223343 ... 0.14660239 0.05923066
   0.40174818]
 [-0.13212799 - 0.3026582 - 0.5194815 \dots -0.08484916 -0.07165129
   0.496635941
 [-0.66055095 -0.3015146 -0.48734954 ... -0.29908112 0.19231087
   0.266074421
 [-0.1676146 -0.13033691 -0.61114514 ... -0.14640707 -0.04662995
   0.4260874 11
```

• Both Word2Vec and FastText are models used to create word embeddings from text data. Word2Vec focuses on capturing word meanings based on their context in sentences, while FastText adds the ability to understand word structure by considering subword information like prefixes and suffixes.

##Train test split

1. Count vectorizer

```
from sklearn.model_selection import train_test_split

# Separate features and target for each matrix
X = X_count
y = data['labeled']

# Split data into train and test sets
X_train_countvec, X_test_countvec, y_train_countvec, y_test_countvec = train_test_split(X, y, test_size=0.2, random_state=42)

# Print the shapes of the training and test sets
```

```
print("X_train_countvec shape:", X_train_countvec.shape)
print("y_train_countvec shape:", y_train_countvec.shape)
print("X_test_countvec shape:", X_test_countvec.shape)
print("y_test_countvec shape:", y_test_countvec.shape)

X_train_countvec shape: (18600, 11975)
y_train_countvec shape: (18600,)
X_test_countvec shape: (4651, 11975)
y_test_countvec shape: (4651,)
```

1. TF-IDF VECTORIZER

```
from sklearn.model_selection import train_test_split

X = X_tfidf
y = data['labeled']

# Split data into train and test sets
X_train_tfidf, X_test_tfidf, y_train_tfidf, y_test_tfidf =
train_test_split(X, y, test_size=0.2, random_state=42)

# Print the shapes of the training and test sets
print("X_train_tfidf shape:", X_train_tfidf.shape)
print("y_train_tfidf shape:", y_train_tfidf.shape)
print("X_test_tfidf shape:", X_test_tfidf.shape)
print("y_test_tfidf shape:", y_test_tfidf.shape)

X_train_tfidf shape: (18600, 11975)
y_train_tfidf shape: (18600,)
X_test_tfidf shape: (4651, 11975)
y_test_tfidf shape: (4651,)
```

MODELLING

BASELINE MODEL

```
from keras.models import Sequential
from keras.layers import Embedding, SimpleRNN, Dense
from keras.callbacks import EarlyStopping

# Define the variables
MAX_NB_WORDS = 1000  # Maximum number of words to consider
EMBEDDING_DIM = 100  # Dimension of the embedding vector
MAX_SEQUENCE_LENGTH = 1000  # Maximum length of the input sequences
epochs = 10
batch_size = 32

#import Libraries
from tensorflow.keras.layers import Embedding, SimpleRNN, Dense
from tensorflow.keras.preprocessing.sequence import pad_sequences #
```