Q1)The electronics data contains a total of [insert total number of rows in el_df] entries. Each entry consists of details such as the reviewer's ID, product ID, review text, overall rating, and other relevant information. This dataset offers valuable insights into customer sentiments and preferences regarding electronic products.

```
    while True:
        chunk = pd.read_json(file, lines=True, nrows=chunk_size)
        if chunk.empty:
            break
        chunks.append(chunk)

# Concatenate all chunks into a single DataFrame
el_df = pd.concat(chunks, ignore_index=True)
print("electonics data loaded into el_df dataframe")
```

```
electonics data loaded into el_df dataframe
```

Q2) Upon examining the metadata, we found that it contains information about product attributes such as product ID, title, brand, price, and features. This dataset allows us to gain a deeper understanding of the characteristics and specifications of the electronic products available.

```
# Initialize an empty list to store chunks of data
chunks = []

# Open the gzipped JSON file and decompress it
with gzip.open('/kaggle/input/information-retrieval/meta_Electronics.json', 'rt', encoding='utf-8') as file:
    # Read the JSON data in chunks
    while True:
        chunk = pd.read_json(file, lines=True, nrows=chunk_size)
        if chunk.empty:
            break
        chunks.append(chunk)

# Concatenate all chunks into a single DataFrame
meta_df = pd.concat(chunks, ignore_index=True)
print("meta data loaded into meta_df dataframe.")
```

```
meta data loaded into meta_df dataframe.
```

Q3)In our analysis of the metadata for electronic products, we focused on items related to headphones. We began by converting the 'title' column to lowercase to ensure consistency in our search. Subsequently, we filtered the dataset to extract entries containing the keywords "headphone" or "headphones" in lowercase, resulting in a dedicated dataframe specifically for headphone products.

Our analysis revealed a total of 27,412 entries within the headphone dataframe. These entries encompass a diverse range of headphone products available in the dataset. By isolating this subset of data, we can perform more targeted analyses and gain insights specific to the headphone market segment.

```python
# Convert the 'title' column to lowercase
meta_df['title_lower'] = meta_df['title'].str.lower()

# Filter dataframe for entries where the title contains "headphone" or "headphones" in lower case
headphone_df = meta_df[meta_df['title_lower'].str.contains('headphone|headphones', na=False)]

# Get the total number of rows for the headphones dataframe
total_rows_headphone = len(headphone_df)
print("3)Total number of rows for the headphone dataframe:", total_rows_headphone)
```

```
3)Total number of rows for the headphone dataframe: 27412
```

```python
average_rating_score=merged_df_hp['overall'].mean()
print("b)average rating score for headphone: ", average_rating_score)
```

```
b)average rating score for headphone:  4.082961309523809
```

```python
num_unique_product=merged_df_hp['asin'].nunique()
print("c)number of unique products for headphone :",num_unique_product)
```

```
c)number of unique products for headphone : 26865
```

```python
bad_rating=merged_df_hp[merged_df_hp['overall']<3]['overall'].count()
print("e) total number of bad rating for the headphone :",bad_rating)
```

```
e) total number of bad rating for the headphone : 1175
```

+ Code   + Markdown

```python
rating_count_series=merged_df_hp['overall'].value_counts()
print("f)number of reviews corrosponding to each rating :-\n", rating_count_series)
```

```
f)number of reviews corrosponding to each rating :-
 overall
5.0    4469
4.0    1629
3.0     791
1.0     659
2.0     516
Name: count, dtype: int64
```

Preprocessing Text Data for Analysis:-

In preparation for text analysis tasks, we executed a comprehensive preprocessing pipeline on the review texts within our dataset. This pipeline involved several steps to ensure the cleanliness and consistency of the text data. Initially, we removed HTML tags and accented characters, ensuring that the text is in a standardized format free from any encoding irregularities. Subsequently, we expanded acronyms using a predefined dictionary, enhancing the readability and interpretability of the text. Furthermore, we eliminated special characters and numbers to focus solely on alphabetic characters, essential for meaningful analysis. Tokenization was then applied to segment the text into individual words, facilitating subsequent processing steps.

Lemmatization and Stopword Removal:-

To enhance the quality of our text data, we employed lemmatization to reduce inflected words to their base or dictionary form. This step helps to consolidate variations of words and improves the coherence of the text corpus. Additionally, we filtered out stopwords, common words that carry little semantic meaning and can skew analysis results. By removing stopwords from our text corpus, we retained only the most relevant words, thereby refining the dataset for subsequent analyses. Finally, the preprocessed text data was saved into a CSV file, ensuring easy access and compatibility with various analytical tools and platforms. This preprocessing pipeline lays the groundwork for insightful text analysis, enabling us to extract meaningful insights and trends from the review data with greater accuracy and efficiency.

```python
def preprocess_text(text):
    # Removing HTML Tags
    text = re.sub(r'<[^>]+>', '', text)

    # Removing accented characters
    text = unicodedata.normalize('NFKD', text).encode('ascii', 'ignore').decode('utf-8', 'ignore')

    # Expanding acronyms
    text = expand_acronyms(text, acronyms_dict)

    # Removing Special Characters and numbers
    text = re.sub(r'[^a-zA-Z\s]', '', text)

    # Tokenization
    tokens = word_tokenize(text)

    # Lemmatization (ensure WordNet corpus is available)
    lemmatizer = WordNetLemmatizer()

    tokens = [lemmatizer.lemmatize(word.lower()) for word in tokens]

    # Removing stopwords
    stop_words = set(stopwords.words('english'))
    tokens = [word for word in tokens if word not in stop_words]
    return tokens

# Drop rows with missing 'reviewText' values
merged_df_hp.dropna(subset=["reviewText"], inplace=True)
```

Sample output:-

```python
for idx, clean_review in enumerate(merged_df_hp['clean_reviewText'].head(10)):
    print(f"Review {idx + 1}: {clean_review}")
```

```
Review 1: ['great', 'headphone', 'cord', 'short']
Review 2: ['im', 'getting', 'listening', 'station', 'going', 'several', 'elementary', 'classroom', 'proved', 'reliable', 'kidproof', 'reasonably', 'priced', 'limit', 'quantity', 'im', 'getting', 'one', 'bran
d', 'brand', 'ive', 'gotten', 'worked']
Review 3: ['suck', 'bought', 'wa', 'walking', 'work', 'one', 'day', 'drug', 'store', 'got', 'ripped', 'paid', 'suck', 'uncomfortable', 'ear', 'soft', 'rubber', 'hard', 'plastic', 'even', 'really', 'molded', 'f
it', 'ear', 'plug', 'wa', 'shorting', 'second', 'plugged', 'requires', 'much', 'twisting', 'bending', 'get', 'sound', 'since', 'dont', 'fit', 'well', 'average', 'ear', 'sound', 'quality', 'end', 'sounding', 's
uper', 'tinny', 'found', 'reverse', 'left', 'right', 'earpiece', 'twist', 'degree', 'angle', 'front', 'sound', 'ok', 'comfortable', 'cant', 'wear', 'like', 'minute', 'white', 'like', 'every', 'earbuds', 'avail
able', 'market', 'today', 'wasnt', 'plugging', 'ipod', 'look', 'yuppie', 'bus', 'suburbia', 'white', 'wire', 'going', 'ear', 'yuppie', 'going', 'start', 'thinking', 'one', 'try', 'trade', 'rare', 'phish', 'boo
tleg', 'going', 'go', 'back', 'using', 'gigantic', 'optimus', 'headcans', 'look', 'silly', 'sound', 'better', 'time', 'spent', 'writing', 'review', 'could', 'panhandled', 'enough', 'cash', 'get', 'better', 'he
adphone', 'maybe', 'stop', 'cheapskate', 'drop', 'money', 'better', 'headphone', 'nah']
Review 4: ['ive', 'using', 'year', 'basically', 'quiet', 'environment', 'sound', 'pretty', 'good', 'arent', 'good', 'similarlypriced', 'studioclass', 'headset', 'youre', 'paying', 'noise', 'cancelling', 'raw',
'capability', 'pretty', 'decent', 'ha', 'acceptable', 'detail', 'treble', 'vocal', 'bass', 'range', 'noise', 'cancelling', 'work', 'pretty', 'well', 'ive', 'taken', 'several', 'intercontinental', 'flight', 'de
cent', 'job', 'blocking', 'engine', 'rumble', 'screaming', 'kid', 'also', 'decent', 'job', 'blocking', 'adult', 'human', 'voice', 'nice', 'office', 'however', 'must', 'pretty', 'big', 'ear', 'ive', 'found', 'f
oam', 'cup', 'arent', 'quite', 'thick', 'enough', 'adequately', 'push', 'driver', 'away', 'outer', 'ear', 'hour', 'use', 'ear', 'start', 'ache', 'bit', 'cord', 'plenty', 'long', 'disconnected', 'middle', 'lin
k', 'optional', 'sony', 'wired', 'remote', 'control', 'adapter', 'convert', 'standard', 'th', 'inch', 'stereo', 'jack', 'dual', 'th', 'inch', 'jack', 'many', 'airplane', 'seat', 'included', 'thing', 'dont', 'l
ike', 'find', 'uncomfortable', 'try', 'lean', 'head', 'window', 'airplane', 'sleep', 'think', 'mdrnoisecancellings', 'might', 'better', 'choice', 'regard', 'theyre', 'also', 'kind', 'bulky', 'unless', 'large',
'carryon', 'keep', 'im', 'paranoid', 'putting', 'seat', 'pocket', 'front', 'since', 'mdrnoisecancellings', 'come', 'soft', 'bag', 'instead', 'rigid', 'clamshell', 'like', 'bose', 'headset', 'havent', 'spill',
'yet', 'active', 'traveller', 'may', 'wish', 'find', 'case', 'thats', 'rigid', 'im', 'pretty', 'happy', 'natural', 'audio', 'sound', 'noise', 'cancelling', 'plenty', 'sufficient', 'price', 'couldnt', 'justif
y', 'bose', 'unless', 'needed', 'protective', 'clamshell', 'ear', 'fit', 'perfectly', 'earcups']
Review 5: ['bought', 'pair', 'headphone', 'replace', 'pair', 'model', 'bought', 'several', 'year', 'ago', 'first', 'pair', 'good', 'sound', 'quality', 'especially', 'fore', 'price', 'nowhere', 'near', 'good',
'highend', 'headphone', 'didnt', 'expect', 'problem', 'wa', 'thin', 'cable', 'eventually', 'broke', 'pair', 'replacement', 'look', 'bear', 'model', 'number', 'comparison', 'end', 'new', 'one', 'terrible', 'dyn
amic', 'range', 'sound', 'flat', 'dull', 'cord', 'ha', 'upgraded', 'much', 'heavier', 'gauge', 'heavy', 'opinion', 'stiff', 'get', 'way', 'pair', 'old', 'new', 'good', 'job', 'blocking', 'background', 'noise',
'really', 'liked', 'original', 'reading', 'painting', 'wanted', 'mute', 'sound', 'rest', 'house', 'new', 'pair', 'still', 'acceptable', 'job', 'blocking', 'sound', 'worth', 'wearing']
Review 6: ['wa', 'looking', 'headphone', 'replace', 'old', 'sony', 'mdrvs', 'losing', 'foam', 'covering', 'wanted', 'limit', 'outside', 'noise', 'leaking', 'sound', 'outsider', 'either', 'need', 'practicing',
'piano', 'without', 'disturbing', 'roommate', 'closedback', 'design', 'seemed', 'favorable', 'reading', 'review', 'headphone', 'trying', 'make', 'selection', 'found', 'major', 'problem', 'closedback', 'desig
n', 'sound', 'resonates', 'distorts', 'making', 'sound', 'like', 'tin', 'ear', 'exepction', 'tried', 'playing', 'keyboard', 'sounded', 'way', 'worse', 'headphone', 'ear', 'bud', 'ever', 'used', 'credit', 'keyb
oard', 'voice', 'piano', 'sounded', 'little', 'better', 'maybe', 'headphone', 'would', 'great', 'listening', 'techno', 'music', 'something', 'certainly', 'classical', 'shall', 'returning']
Review 7: ['fine', 'sound', 'good', 'reception', 'decent', 'range', 'look', 'like', 'get', 'around', 'yard', 'without', 'making', 'big', 'complaint', 'comfortable', 'would', 'say', 'fidelity', 'slightly', 'l
e', 'earbuds', 'im', 'used', 'separation', 'better', 'expected', 'fair', 'amount', 'hiss', 'music', 'im', 'pretty', 'picky', 'think', 'totally', 'tolerable']
Review 8: ['headset', 'inexpensive', 'dont', 'expect', 'product', 'year', 'come', 'however', 'ha', 'good', 'sound', 'ear', 'foam', 'denser', 'low', 'end', 'product', 'make', 'comfortable']
Review 9: ['advertised', 'would', 'gladly', 'transact', 'business', 'seller']
Review 10: ['really', 'satisfied', 'purchase', 'headphone', 'clear', 'even', 'pick', 'signal', 'great', 'distance', 'promised', 'problem', 'large', 'size', 'head', 'phone', 'still', 'worth', 'also', 'wa', 'abl
e', 'tune', 'fm', 'radio', 'station', 'get', 'tired', 'listening', 'music', 'laptop', 'tune', 'radio', 'station', 'great', 'music', 'well', 'done', 'jvc']
```

Output:-

```python
# Display the top 20 most reviewed brands
print("Top 20 most reviewed brands:")
print(top_20_most_reviewed_brands)

# Display the top 20 least reviewed brands
print("\nTop 20 least reviewed brands:")
print(top_20_least_reviewed_brands)
```

```
Top 20 most reviewed brands:
                 brand  review_count
14590             Sony        160506
9609          Logitech        140225
13902          Samsung        139391
13908          SanDisk        133870
1091      AmazonBasics        121316
3208             Canon         93409
2100            Belkin         81683
1488              Asus         70776
17268  Western Digital         60103
11211            Nikon         59182
14812         StarTech         55908
1324             Apple         53166
3055     Cable Matters         49848
10902          NETGEAR         49500
12020        Panasonic         47744
6505            Garmin         47741
15369          TP-LINK         46937
13840          Sabrent         45073
1217             Anker         44123
10305      Mediabridge         43899
```

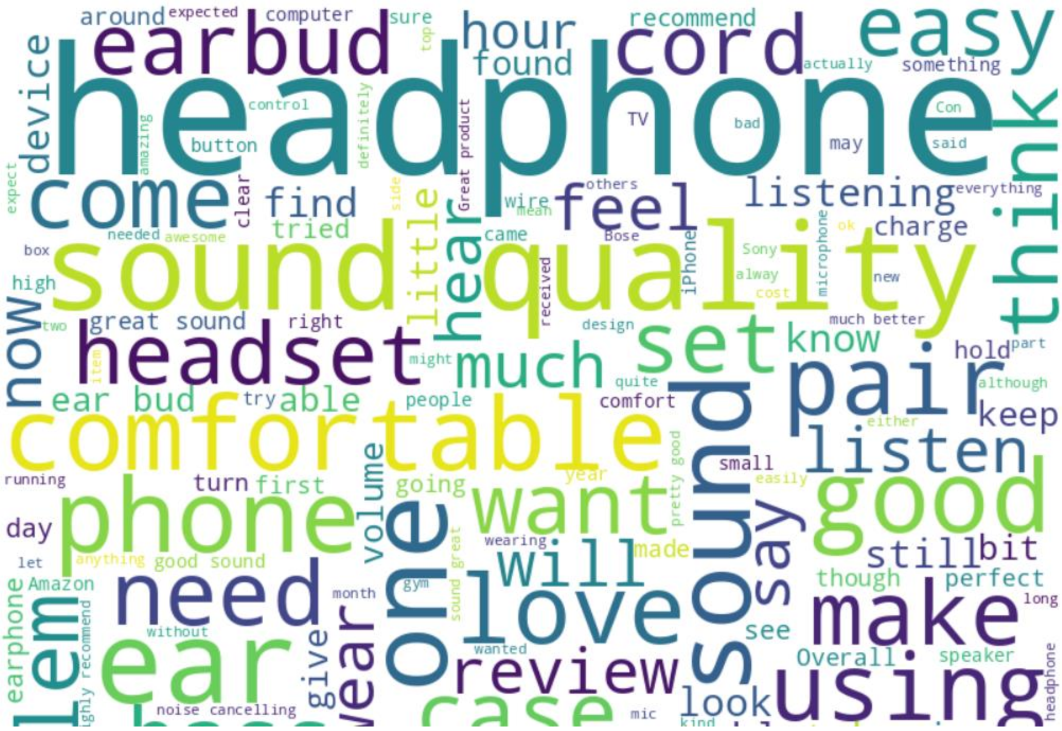
```
Top 20 least reviewed brands:
                                  brand  review_count
14807                         Star Toys             1
4493                    Digital Antenna             1
10017             MULTI-REGION DVD PLAYER           1
3687                   Concept Lighting             2
7540                              Honda             2
8984                           Kwik Tek             2
10421            Michael Pearl; Debi Pearl          2
13007  Radio Shack Corporation Radio Shack          2
4362                         Deer River             2
3949                         Cybersnipa             2
12546                       Prima Cases             2
1382                           Areaware             3
15893                           ToolUSA             3
16686                          ValuSoft             3
9424            Leslie Dame Enterprises             3
8708                          Kashimura             3
16640                              VRUM             3
10954                      NOIZY Brands             3
12322                          PixiModo             3
922                            Air King             3
```

Visualizing Review Sentiments

Utilizing WordClouds, we visualize the most frequent terms within reviews associated with 'Good' and 'Bad' ratings, offering a concise snapshot of sentiments expressed by customers.

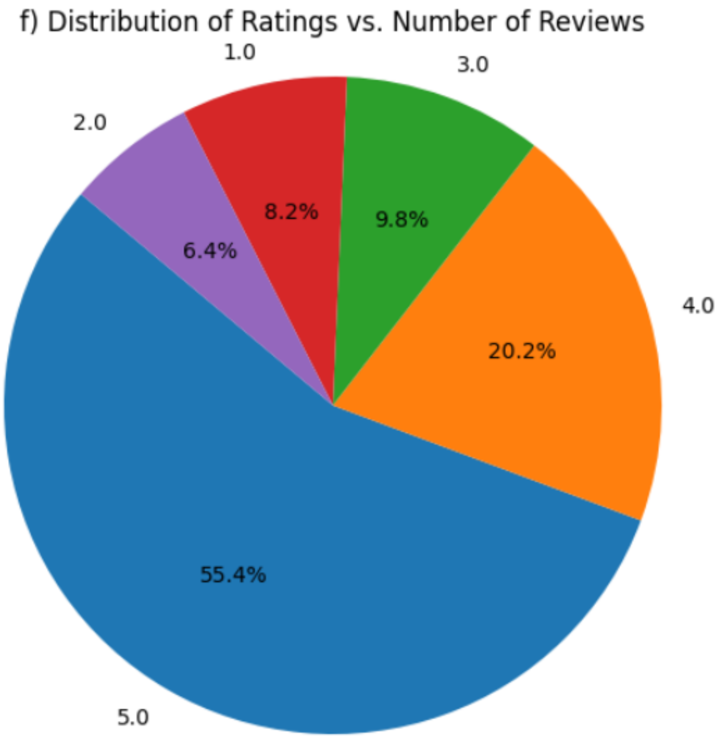
Good rating :-

Word Cloud for Good Ratings



Bad rating :-

Word Cloud for Bad Ratings




```python
# Extract words and their frequencies from the WordCloud for 'Good' ratings
good_word_freq = good_wordcloud.words_

# Extract words and their frequencies from the WordCloud for 'Bad' ratings
bad_word_freq = bad_wordcloud.words_

# Report the most commonly used words for positive reviews
print("Most commonly used words for positive reviews:")
for word, freq in good_word_freq.items():
    print(f"{word}: {freq}")

# Report the most commonly used words for negative reviews
print("\nMost commonly used words for negative reviews:")
for word, freq in bad_word_freq.items():
    print(f"{word}: {freq}")
```

```
Most commonly used words for positive reviews:
headphone: 1.0
sound: 0.6006768189509306
ear: 0.42455006922012
use: 0.39732348869404704
good: 0.3913244116289802
one: 0.36548223350253806
great: 0.33902476542070453
music: 0.2910321489001692
bass: 0.2578064913090294
phone: 0.24180895246885095
fit: 0.23934779264728503
well: 0.22934933087217352
work: 0.22504230118443316
earbud: 0.22504230118443316
sound quality: 0.2248884786955853
comfortable: 0.22011998154130133
pair: 0.21658206429780033
will: 0.21227503461006
price: 0.21181356714351637
```

```python
# Extract words and their frequencies from the WordCloud for 'Good' ratings
good_word_freq = good_wordcloud.words_

# Extract words and their frequencies from the WordCloud for 'Bad' ratings
bad_word_freq = bad_wordcloud.words_

# Report the most commonly used words for positive reviews
print("Most commonly used words for positive reviews:")
for word, freq in good_word_freq.items():
    print(f"{word}: {freq}")

# Report the most commonly used words for negative reviews
print("\nMost commonly used words for negative reviews:")
for word, freq in bad_word_freq.items():
    print(f"{word}: {freq}")
```

```
Most commonly used words for positive reviews:
headphone: 1.0
sound: 0.6006768189509306
ear: 0.42455006922012
use: 0.39732348869404704
good: 0.3913244116289802
one: 0.36548223350253806
great: 0.33902476542070453
music: 0.2910321489001692
bass: 0.2578064913090294
phone: 0.24180895246885095
fit: 0.23934779264728503
well: 0.22934933087217352
work: 0.22504230118443316
earbud: 0.22504230118443316
sound quality: 0.2248884786955853
comfortable: 0.22011998154130133
pair: 0.21658206429780033
will: 0.21227503461006
price: 0.21181356714351637
```

```
Most commonly used words for negative reviews:
headphone: 1.0
sound: 0.8691588785046729
ear: 0.49182242990654207
one: 0.4380841121495327
good: 0.42757009345794394
work: 0.3901869158878505
use: 0.3878504672897196
product: 0.30257009345794394
will: 0.27686915887850466
even: 0.2488317757009346
time: 0.24766355140186916
phone: 0.24766355140186916
pair: 0.23598130841121495
really: 0.23014018691588786
better: 0.2207943925233645
great: 0.2207943925233645
review: 0.2161214953271028
earbud: 0.21495327102803738
fit: 0.21378504672897197
bass: 0.20677570093457945
```

```
    accuracy                           0.81      2016
   macro avg       0.64      0.48      0.51      2016
weighted avg       0.77      0.81      0.77      2016

Training Random Forest Classifier...
Evaluating Random Forest Classifier...
              precision    recall  f1-score   support

     Average       0.33      0.01      0.02       189
         Bad       0.85      0.26      0.40       297
        Good       0.79      0.99      0.88      1530

    accuracy                           0.79      2016
   macro avg       0.66      0.42      0.43      2016
weighted avg       0.76      0.79      0.73      2016

Training Support Vector Classifier...
Evaluating Support Vector Classifier...
              precision    recall  f1-score   support

     Average       0.38      0.02      0.03       189
         Bad       0.81      0.32      0.46       297
        Good       0.80      0.99      0.89      1530

    accuracy                           0.80      2016
   macro avg       0.66      0.44      0.46      2016
weighted avg       0.76      0.80      0.75      2016

Training Decision Tree Classifier...
Evaluating Decision Tree Classifier...
              precision    recall  f1-score   support

     Average       0.16      0.15      0.15       189
         Bad       0.45      0.44      0.45       297
        Good       0.84      0.85      0.84      1530

    accuracy                           0.72      2016
   macro avg       0.48      0.48      0.48      2016
weighted avg       0.72      0.72      0.72      2016
```

```python
#12 top 10 product
# Group by 'asin' (product ID) and sum up the ratings for each product
product_sum_ratings = merged_df.groupby('asin')['overall'].sum()

# Sort the products by sum ratings in descending order
top_10_products = product_sum_ratings.sort_values(ascending=False).head(10)

# Print the top 10 products by user sum ratings
print("Top 10 Products by User Sum Ratings:")
for i, (product_id, sum_ratings) in enumerate(top_10_products.items(), 1):
    print(f"{i}. Product ID: {product_id}, Sum Ratings: {sum_ratings}")
```

```
Top 10 Products by User Sum Ratings:
1. Product ID: B003L1ZYYW, Sum Ratings: 41258.0
2. Product ID: B00004ZCJJ, Sum Ratings: 40144.0
3. Product ID: B00004ZCJI, Sum Ratings: 40144.0
4. Product ID: B00009KLAE, Sum Ratings: 40124.0
5. Product ID: B0019HL8Q8, Sum Ratings: 38880.0
6. Product ID: B0019EHU8G, Sum Ratings: 37022.0
7. Product ID: B0015DYMVO, Sum Ratings: 31546.0
8. Product ID: B000VS4HDM, Sum Ratings: 31544.0
9. Product ID: B00M55C0NS, Sum Ratings: 29391.0
10. Product ID: B000BQ7GW8, Sum Ratings: 28209.0
```