


Data Collection and Preprocessing Phase

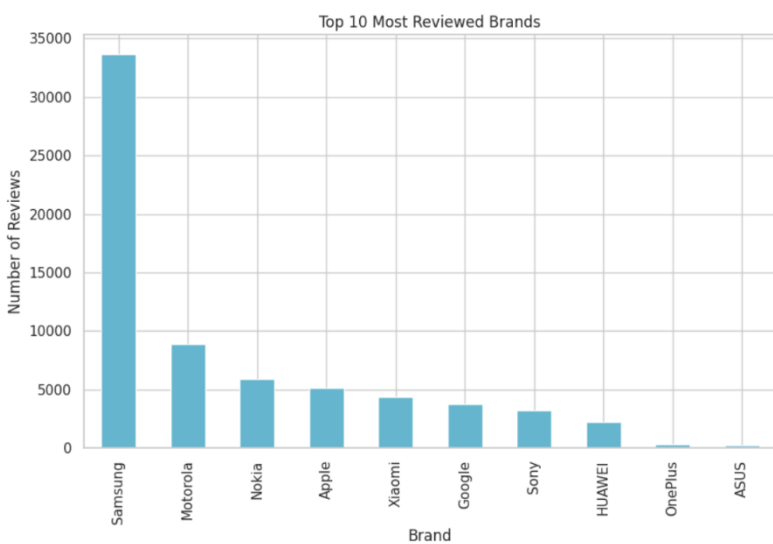
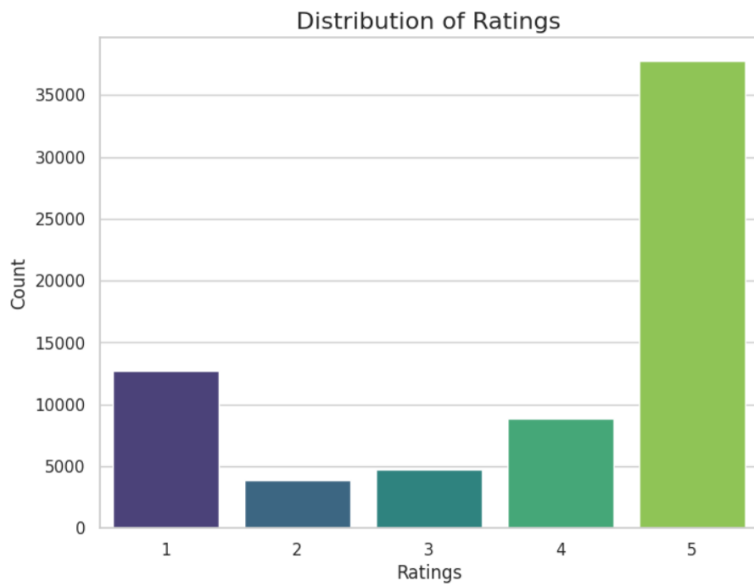
Date	03 October 2024
Team ID	LTVIP2024TMID24974
Project Title	Analysis Of Amazon Cell Phone Reviews
Maximum Marks	6 Marks

Data Exploration and Preprocessing Report

The data exploration involves understanding the dataset's structure, identifying missing values, and visualizing key aspects like rating distribution and review lengths. Preprocessing includes cleaning the text (lowercasing, removing special characters, and stopwords), tokenization, and lemmatization. Additionally, sentiment labeling is applied to categorize reviews as positive, negative, or neutral. The result is a clean, structured dataset ready for further analysis, such as sentiment analysis, rating prediction, or generating insights from customer reviews.

Section	Description																																																						
Data Overview	<u>Dimension:</u> 67986 rows × 17 columns																																																						
	<u>Descriptive statistics:</u>																																																						
	<pre>[] print(merged_df.describe())</pre>																																																						
																																																							
	<table><tr><td></td><td>rating_x</td><td>helpfulVotes</td><td>rating_y</td><td>totalReviews</td><td>price \</td></tr><tr><td>count</td><td>67986.000000</td><td>27215.000000</td><td>67986.000000</td><td>67986.000000</td><td>67986.000000</td></tr><tr><td>mean</td><td>3.807916</td><td>8.229690</td><td>3.766826</td><td>373.742800</td><td>222.050506</td></tr><tr><td>std</td><td>1.582906</td><td>31.954877</td><td>0.429197</td><td>262.560876</td><td>188.863986</td></tr><tr><td>min</td><td>1.000000</td><td>1.000000</td><td>1.000000</td><td>1.000000</td><td>0.000000</td></tr><tr><td>25%</td><td>3.000000</td><td>1.000000</td><td>3.500000</td><td>153.000000</td><td>103.980000</td></tr><tr><td>50%</td><td>5.000000</td><td>2.000000</td><td>3.800000</td><td>336.000000</td><td>179.990000</td></tr><tr><td>75%</td><td>5.000000</td><td>5.000000</td><td>4.100000</td><td>558.000000</td><td>300.550000</td></tr><tr><td>max</td><td>5.000000</td><td>990.000000</td><td>5.000000</td><td>983.000000</td><td>999.990000</td></tr></table>		rating_x	helpfulVotes	rating_y	totalReviews	price \	count	67986.000000	27215.000000	67986.000000	67986.000000	67986.000000	mean	3.807916	8.229690	3.766826	373.742800	222.050506	std	1.582906	31.954877	0.429197	262.560876	188.863986	min	1.000000	1.000000	1.000000	1.000000	0.000000	25%	3.000000	1.000000	3.500000	153.000000	103.980000	50%	5.000000	2.000000	3.800000	336.000000	179.990000	75%	5.000000	5.000000	4.100000	558.000000	300.550000	max	5.000000	990.000000	5.000000	983.000000	999.990000
		rating_x	helpfulVotes	rating_y	totalReviews	price \																																																	
	count	67986.000000	27215.000000	67986.000000	67986.000000	67986.000000																																																	
	mean	3.807916	8.229690	3.766826	373.742800	222.050506																																																	
	std	1.582906	31.954877	0.429197	262.560876	188.863986																																																	
	min	1.000000	1.000000	1.000000	1.000000	0.000000																																																	
25%	3.000000	1.000000	3.500000	153.000000	103.980000																																																		
50%	5.000000	2.000000	3.800000	336.000000	179.990000																																																		
75%	5.000000	5.000000	4.100000	558.000000	300.550000																																																		
max	5.000000	990.000000	5.000000	983.000000	999.990000																																																		
<table><tr><td></td><td>originalPrice</td></tr><tr><td>count</td><td>67986.000000</td></tr><tr><td>mean</td><td>84.057634</td></tr><tr><td>std</td><td>201.923373</td></tr><tr><td>min</td><td>0.000000</td></tr><tr><td>25%</td><td>0.000000</td></tr><tr><td>50%</td><td>0.000000</td></tr><tr><td>75%</td><td>0.000000</td></tr><tr><td>max</td><td>999.990000</td></tr></table>		originalPrice	count	67986.000000	mean	84.057634	std	201.923373	min	0.000000	25%	0.000000	50%	0.000000	75%	0.000000	max	999.990000																																					
	originalPrice																																																						
count	67986.000000																																																						
mean	84.057634																																																						
std	201.923373																																																						
min	0.000000																																																						
25%	0.000000																																																						
50%	0.000000																																																						
75%	0.000000																																																						
max	999.990000																																																						

Univariate Analysis



Bivariate Analysis

—

Multivariate Analysis



Data Preprocessing Code Screenshots

Loading Data

LOADING THE DATASET

```

# Load the datasets (replace the file paths with your actual file paths)
items_df = pd.read_csv('/content/drive/MyDrive/20191226-items.csv')
reviews_df = pd.read_csv('/content/drive/MyDrive/20191226-reviews.csv')

# Merge datasets based on 'asin' to combine product info with reviews
merged_df = pd.merge(reviews_df, items_df, on='asin')

# View the first few rows to check the loaded data
print(merged_df.head())
  
```

Handling Missing Data

```

▶ print(merged_df.isnull().sum())

⇒
asin          0
name          3
rating_x      0
date          0
verified      0
title_x       29
body          26
helpfulVotes  40771
brand         200
title_y       0
url           0
image         0
rating_y      0
reviewUrl     0
totalReviews  0
price         0
originalPrice 0
dtype: int64

```

Data Preprocessing

TEXT PREPROCESSING

```

▶ # Define a function to preprocess the review text
def preprocess_text(text):
    # Check if the text is a string
    if isinstance(text, str):
        # Convert to lowercase
        text = text.lower()
        # Remove punctuation using NLTK's tokenizer
        text = nltk.RegexpTokenizer(r'\w+').tokenize(text)
        # Re-join words into a single string
        return ' '.join(text)
    else:
        # Handle non-string values (e.g., return an empty string or a placeholder)
        return ''

# Apply preprocessing to the review body column
merged_df['cleaned_review'] = merged_df['body'].apply(preprocess_text)

# View the preprocessed reviews
print(merged_df[['body', 'cleaned_review']].head())



```

```

⇒
body \
0 I had the Samsung A600 for awhile which is abs...
1 Due to a software issue between Nokia and Spri...
2 This is a great, reliable phone. I also purcha...
3 I love the phone and all, because I really did...
4 The phone has been great for every purpose it ...

cleaned_review
0 i had the samsung a600 for awhile which is abs...
1 due to a software issue between nokia and spri...

```

<h2>Visualizations</h2>	<pre>[] import matplotlib.pyplot as plt import seaborn as sns # Set plot style sns.set(style="whitegrid")</pre> <p>  # Plot the distribution of ratings plt.figure(figsize=(8, 6)) sns.countplot(x='rating_x', data=merged_df, palette='viridis') plt.title('Distribution of Ratings', fontsize=16) plt.xlabel('Ratings', fontsize=12) plt.ylabel('Count', fontsize=12) plt.show() </p> <p>  <ipython-input-11-32ceca067aaa>3: FutureWarning: Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'x' variable to 'hue' and set 'legend=False' for the same effect. sns.countplot(x='rating_x', data=merged_df, palette='viridis') </p> 
<h2>Save Processed Data</h2>	<hr/>