# Data Collection and Preprocessing Phase

| | |
|---|---|
| Date | 15 March 2024 |
| Team ID | SWTID1727420425 |
| Project Title | Analysis of amazon cell phone reviews using nlp technique |
| Maximum Marks | 6 Marks |

## Data Exploration and Pre processing Report

The Amazon cell phone review analysis involves two datasets: item metadata and customer reviews. Initial exploration revealed insights into review counts, ratings distribution, and product coverage. Data pre processing included handling missing values, cleaning review text, and converting review dates for trend analysis. Sentiment labels (positive, neutral, negative) were derived from ratings, and features like review length were added. This prepared data is now clean and ready for advanced NLP tasks like sentiment analysis and topic modeling.

| **Section** | **Description** |
|---|---|
| Data Overview | The project utilizes two datasets for analyzing Amazon cell phone reviews: one containing product metadata (e.g., name, brand, price) and another with customer reviews (e.g., ratings, review text, review date). The review dataset provides insights into customer feedback, while the item dataset complements it with product details. Together, these datasets enable the application of NLP techniques to understand customer sentiments, trends, and preferences, forming the foundation for comprehensive analysis. |
| Resizing | In the context of this project, resizing involves standardizing the size and structure of review text for efficient NLP processing. Long reviews were truncated or summarized to a manageable length, while very short reviews were either padded or excluded, depending on their relevance. This step ensures uniformity across the dataset, improving the performance and accuracy of NLP models by focusing on meaningful and consistent input data. |

| | |
|---|---|
| Normalization | Normalization in this project focuses on transforming the review text into a consistent format for NLP analysis. This includes converting all text to lowercase, removing special characters, punctuation, and numbers, and standardizing spellings. Additionally, lemmatization was applied to reduce words to their base forms, ensuring uniformity while preserving meaning. These steps help eliminate noise and improve the accuracy of NLP techniques such as sentiment analysis and topic modeling. |
| Edge Detection | Edge detection is not directly applicable to textual data analysis in this project, as it is a technique primarily used in image processing to identify object boundaries. However, in the context of NLP, analogous techniques might involve identifying "edges" of significant meaning, such as detecting key phrases, transitions in sentiment, or abrupt shifts in topic within reviews, which can be explored through advanced methods like keyword extraction or topic segmentation. |
| Color Space Conversion | Color space conversion is not directly relevant to text-based NLP tasks in this project, as it pertains to image processing and the transformation of pixel data between color models (e.g., RGB to grayscale). For this project, the focus remains on textual data processing, where equivalent steps involve text cleaning and transformation rather than handling visual data. |
| **Data Preprocessing Code Screenshots** | |
| Loading Data | |

| | |
|---|---|
| | + Code + Text<br>• Data Collection<br>`[ ] #load both items and review data`<br>`items = pd.read_csv('/content/20191226-items.csv')`<br>`reviews = pd.read_csv('/content/20191226-reviews.csv')`<br>• Checking the structure of dataset<br>`[ ] # Check column names in each file`<br>`print("Columns in reviews:", reviews.columns)`<br>`print("Columns in items:", items.columns)`<br>`Columns in reviews: Index(['asin', 'name', 'rating', 'date', 'verified', 'title', 'body', 'helpfulVotes'], dtype='object')`<br>`Columns in items: Index(['asin', 'brand', 'title', 'url', 'image', 'rating', 'reviewUrl', 'totalReviews', 'price', 'originalPrice'], dtype='object')` |
| Merging the Data | • Merging the data<br>`[ ] # Merge the datasets on a common column (e.g., 'product_id')`<br>`merged_data = pd.merge(reviews, items, on='asin', how='left')`<br>`[ ] # Inspect the merged dataset`<br>`print("Merged Data Shape:", merged_data.shape)`<br>`print("Columns in Merged Data:", merged_data.columns)`<br>`Merged Data Shape: (67986, 17)`<br>`Columns in Merged Data: Index(['asin', 'name', 'rating_x', 'date', 'verified', 'title_x', 'body', 'helpfulVotes', 'brand', 'title_y', 'url', 'image', 'rating_y', 'reviewUrl', 'totalReviews', 'price', 'originalPrice'], dtype='object')` |
| Handling missing data | • handling missing data<br>`[ ] features = ['rating_x', 'verified', 'title_x', 'body', 'brand', 'price', 'originalPrice']`<br>`target = 'helpfulVotes'`<br>`[ ] # Keep only necessary columns and drop rows with missing values`<br>`filtered_data = merged_data[features + [target]].dropna()`<br>`[ ] print("Filtered Data Shape:", filtered_data.shape)`<br>`Filtered Data Shape: (27069, 8)` |
| Data Exploration | • Data exploration<br>`merged_data.head()`<br>  |

```
merged_data.describe()
```

| | rating_x | helpfulVotes | rating_y | totalReviews | price | originalPrice |
|---|---|---|---|---|---|---|
| count | 67986.000000 | 27215.000000 | 67986.000000 | 67986.000000 | 67986.000000 | 67986.000000 |
| mean | 3.807916 | 8.229690 | 3.766826 | 373.742800 | 222.050506 | 84.057634 |
| std | 1.582906 | 31.954877 | 0.429197 | 262.560876 | 188.863986 | 201.923373 |
| min | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 0.000000 | 0.000000 |
| 25% | 3.000000 | 1.000000 | 3.500000 | 153.000000 | 103.980000 | 0.000000 |
| 50% | 5.000000 | 2.000000 | 3.800000 | 336.000000 | 179.990000 | 0.000000 |
| 75% | 5.000000 | 5.000000 | 4.100000 | 558.000000 | 300.550000 | 0.000000 |
| max | 5.000000 | 990.000000 | 5.000000 | 983.000000 | 999.990000 | 999.990000 |

```
# Check for null values
merged_data.isnull().sum()
```

| | 0 |
|---|---|
| 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

Splitting the data

```
5. Splitting the data

[ ] # Split the data for training and validation
    X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.2, random_state=42)
```