Amazon Product Reviews Analysis Project

2. Project Overview

Dataset Description

The project utilizes the Amazon Product Reviews dataset from Kaggle, which contains customer reviews for various Amazon products. The dataset includes key information such as: - Product names and categories - Customer reviews and ratings (1-5 stars) - Review text and metadata - Product categories and descriptions

Initial Hypothesis

1. Product reviews can be effectively categorized into distinct sentiment classes (positive, negative, neutral) based on both star ratings and review text.
2. Products can be meaningfully clustered into 4-6 meta-categories based on their descriptions and reviews.
3. Review sentiment patterns within product categories can reveal valuable insights about product performance and customer satisfaction.

Analysis Structure and Process

1. **Data Loading and Initial Assessment**
   - Loaded the dataset using pandas
   - Performed initial data exploration and statistics
   - Identified key columns and data types

```
df = pd.read_csv("dataset.csv")
df.columns
df['reviews.rating'].value_counts()
```

2. **Feature Engineering**
   - Mapped star ratings to sentiment classes:
     - 1-2 stars → Negative
     - 3 stars → Neutral
     - 4-5 stars → Positive
   - Created text features for clustering
   - Generated product embeddings using SentenceTransformer
3. **Model Development**
   - Implemented sentiment analysis using DistilBERT
   - Developed product clustering using K-means
   - Created summarization pipeline using BART

3. Data Wrangling and Cleaning

Major Challenges

1. **Missing Data**

```
df.isnull().sum()
```

   - Handled missing review texts

- Dealt with missing product categories
- Managed incomplete ratings

2. **Duplicate Reviews**

```
unique_count = df['name'].nunique()
duplicate_count = df['name'].duplicated().sum()
```

- Identified and removed duplicate reviews
- Preserved unique product information
- Maintained data integrity

3. **Text Preprocessing**

```
def preprocess_text(text):
    # Lowercase
    text = text.lower()
    # Remove HTML tags
    text = re.sub(r'<.*?>', '', text)
    # Remove URLs
    text = re.sub(r"http\S+|www\S+|https\S+", '', text)
    # Remove user mentions and hashtags
    text = re.sub(r'\@\w+|\#', '', text)
    # Remove punctuation
    text = text.translate(str.maketrans('', '', string.punctuation))
    # Remove numbers
    text = re.sub(r'\d+', '', text)
    # Remove extra whitespace
    text = re.sub(r'\s+', ' ', text).strip()
    return text
```

Data Enrichment Methods

1. **Text Feature Extraction**

- Used SentenceTransformer for generating embeddings
- Created TF-IDF features for text analysis

```
model = SentenceTransformer('all-MiniLM-L6-v2')
embeddings = model.encode(product_texts, show_progress_bar=True)
```

2. **Category Clustering**

```
kmeans = KMeans(n_clusters=num_clusters, random_state=42)
clusters = kmeans.fit_predict(embeddings)
```

- Implemented K-means clustering
- Generated cluster labels using most common words
- Created meaningful category names

3. **Sentiment Analysis Enhancement**

- Used TextBlob for additional sentiment features
- Extracted pros and cons from reviews

```python
def analyze_sentiment(reviews):
    pros = defaultdict(int)
    cons = defaultdict(int)
    for review in reviews:
        blob = TextBlob(review)
        for sentence in blob.sentences:
            polarity = sentence.sentiment.polarity
            # ... sentiment analysis logic ...
```

Challenge Resolution

1. **Data Quality Improvements**
    - Implemented robust text preprocessing
    - Handled multilingual reviews
    - Standardized product categories
2. **Feature Engineering Solutions**
    - Created meaningful product clusters
    - Generated sentiment scores
    - Extracted key product attributes
3. **Performance Optimization**
    - Used efficient data structures
    - Implemented batch processing
    - Optimized memory usage

The data cleaning and preprocessing steps were crucial for ensuring high-quality input for our models and generating meaningful insights from the reviews.

4. Exploratory Data Analysis

Analysis Methods

1. **Distribution Analysis**

```python
# Rating Distribution
sns.set_theme(style='darkgrid', font_scale=1.15, palette="Set3")
ax = sns.countplot(x='reviews.rating', data=df)
for p in ax.patches:
    ax.annotate('{}'.format(p.get_height()), (p.get_x()+0.1, p.get_height()+50))
```

2. **Product Category Analysis**

```python
# Product Category Distribution
plt.figure(figsize=(10, 6))
scatter = plt.scatter(reduced_embeddings[:, 0], reduced_embeddings[:, 1],
                     c=clusters, cmap='viridis', alpha=0.6)
plt.title('Product Category Clustering')
plt.colorbar(scatter)
```

3. **Sentiment Patterns**

- Analyzed sentiment distribution across categories
- Investigated rating patterns over time
- Examined correlation between review length and sentiment

Key Insights

1. **Rating Distribution**
    - Most reviews are positively skewed (4-5 stars)
    - Small percentage of extremely negative reviews (1-2 stars)
    - Neutral reviews (3 stars) are relatively uncommon
2. **Category Patterns**
    - Identified 4 distinct product clusters
    - Electronics and accessories form the largest category
    - Some categories show more polarized reviews than others
3. **Review Characteristics**
    - Longer reviews tend to be more detailed and balanced
    - Negative reviews often contain more specific details
    - Positive reviews frequently use common positive phrases

5. Teamwork & Project Management

Workflow Execution

1. **Original Plan vs. Reality**
    - Successfully implemented core functionality
    - Added additional features:
        - Enhanced visualization
        - More sophisticated clustering
        - Improved sentiment analysis
2. **Timeline Management**
    - Met most deadlines
    - Some delays in model optimization
    - Successfully adapted to challenges

Team Collaboration

1. **Strengths**
    - Clear communication
    - Regular code reviews
    - Effective task distribution
2. **Areas for Improvement**
    - Better documentation practices
    - More frequent progress updates
    - Enhanced version control practices

Risk Management

1. **Identified Risks**
    - Data quality issues
    - Model performance
    - Technical dependencies
2. **Mitigation Strategies**
    - Regular data validation
    - Model performance monitoring
    - Backup solutions for critical components

6. Major Obstacles

Primary Challenges

1. **Model Performance Issues**
    - Initial sentiment analysis accuracy was lower than expected
    - Clustering results needed refinement
    - Resource constraints with large models
2. **Resolution Steps**
    - Implemented more sophisticated preprocessing
    - Used lighter, more efficient models
    - Enhanced feature engineering

Lessons Learned

1. **Technical Insights**
    - Importance of thorough data preprocessing
    - Value of efficient model selection
    - Need for robust error handling
2. **Process Improvements**
    - Better initial planning
    - More comprehensive testing
    - Regular performance monitoring

Future Improvements

1. **Potential Enhancements**
    - Implement more sophisticated clustering
    - Add multi-language support
    - Enhance visualization capabilities
2. **Alternative Approaches**
    - Consider different model architectures
    - Explore additional data sources
    - Implement more automated testing

7. Conclusion and Insights

Hypothesis Evaluation

1. **Sentiment Classification**
   - Successfully implemented three-class sentiment analysis
   - Achieved good accuracy in rating prediction
   - Validated correlation between ratings and review text
2. **Category Clustering**
   - Successfully identified meaningful product categories
   - Clustering provided useful insights
   - Some categories showed distinct review patterns

Key Findings

1. **Product Insights**
   - Identified key factors in product satisfaction
   - Discovered common complaint patterns
   - Found correlation between price and review sentiment
2. **Review Patterns**
   - Longer reviews tend to be more informative
   - Sentiment patterns vary by category
   - Time-based trends in review behavior

Implications

1. **Business Impact**
   - Improved product categorization
   - Better understanding of customer satisfaction
   - Identified areas for product improvement
2. **Technical Implications**
   - Demonstrated effectiveness of transformer models
   - Validated clustering approach
   - Identified areas for model improvement

Unanswered Questions

1. **Technical Limitations**
   - Impact of review age on relevance
   - Cross-category product relationships
   - Long-term sentiment trends
2. **Future Research**
   - Multi-language review analysis
   - Deep dive into specific categories
   - Time-series analysis of ratings

The project successfully demonstrated the value of NLP in analyzing product reviews, while also highlighting areas for future improvement and research.