@Sahar Sheshah
sheshah45@gmail.com

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.

Data Wrangling and Cleaning

Major Challenges: -

1. **Missing Data** → solve

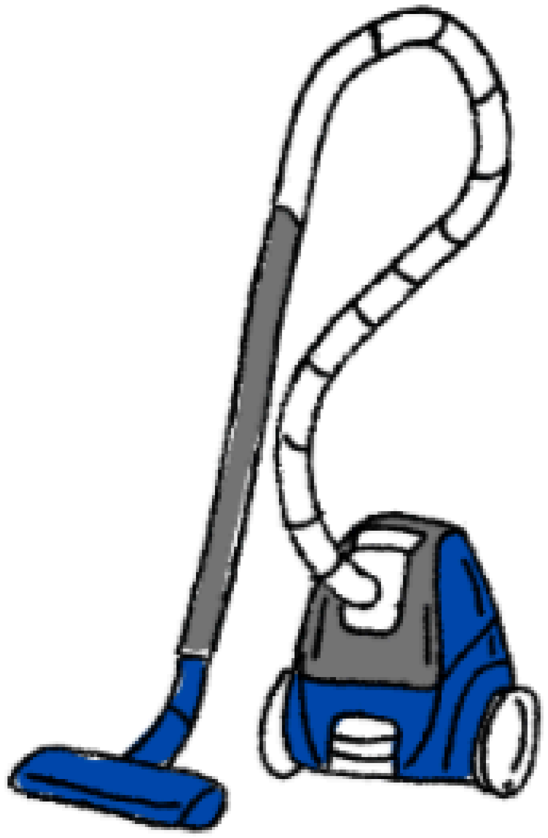
```
df.isnull().sum()
```

2. **Duplicate Reviews** → solve

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

3. **Text Preprocessing** → solve

```
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**
2. **Category Clustering**
3. **Sentiment Analysis Enhancement**

> ***"Used SentenceTransformer***
> ***Created TF-IDF features***
>
> ***Implemented K-means clustering***
>
> ***Used TextBlob***

Challenge Resolution: -

1. Data Quality Improvements

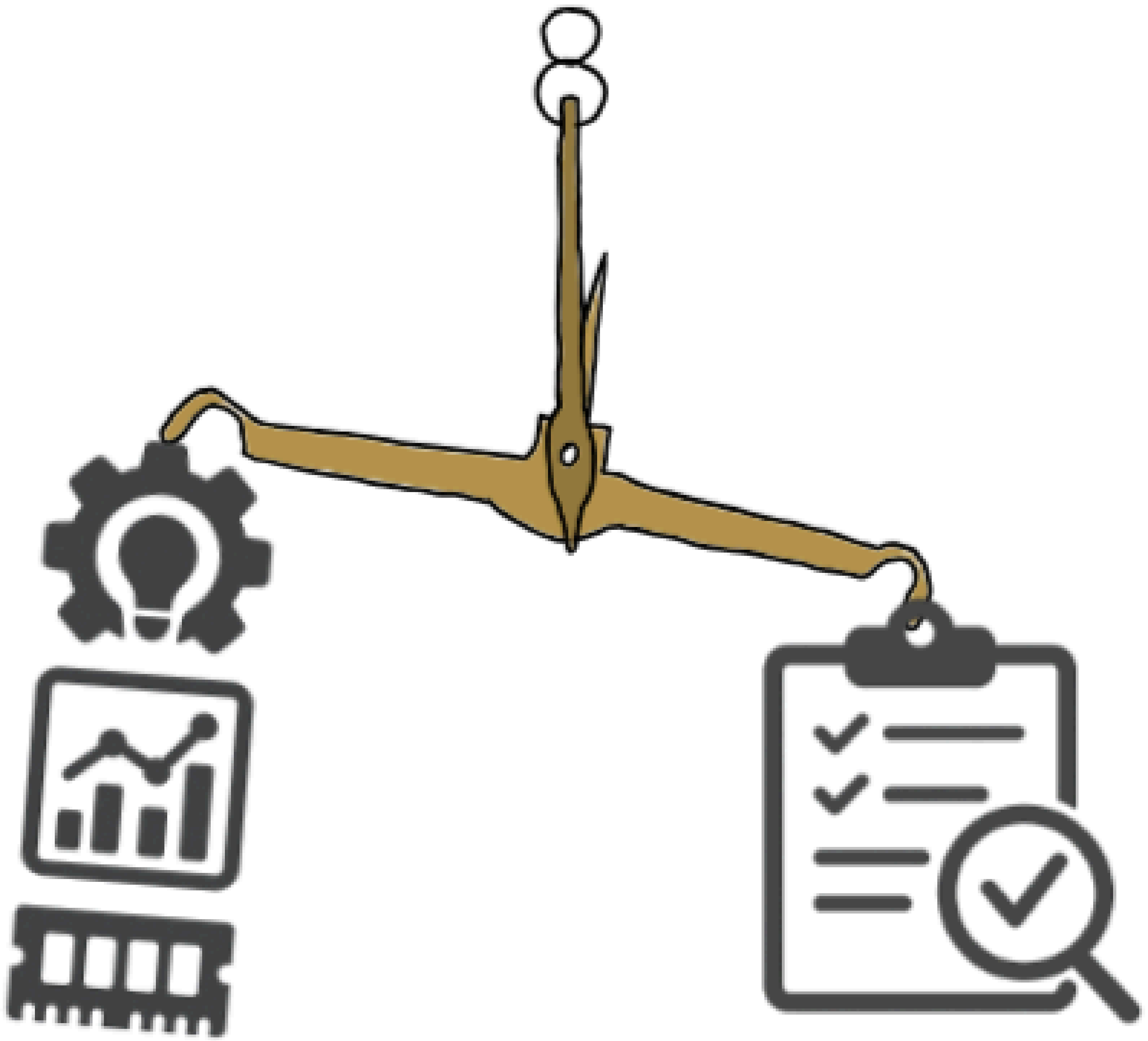
- Implemented robust text preprocessing
- Handled multilingual reviews
- Standardized product categories

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

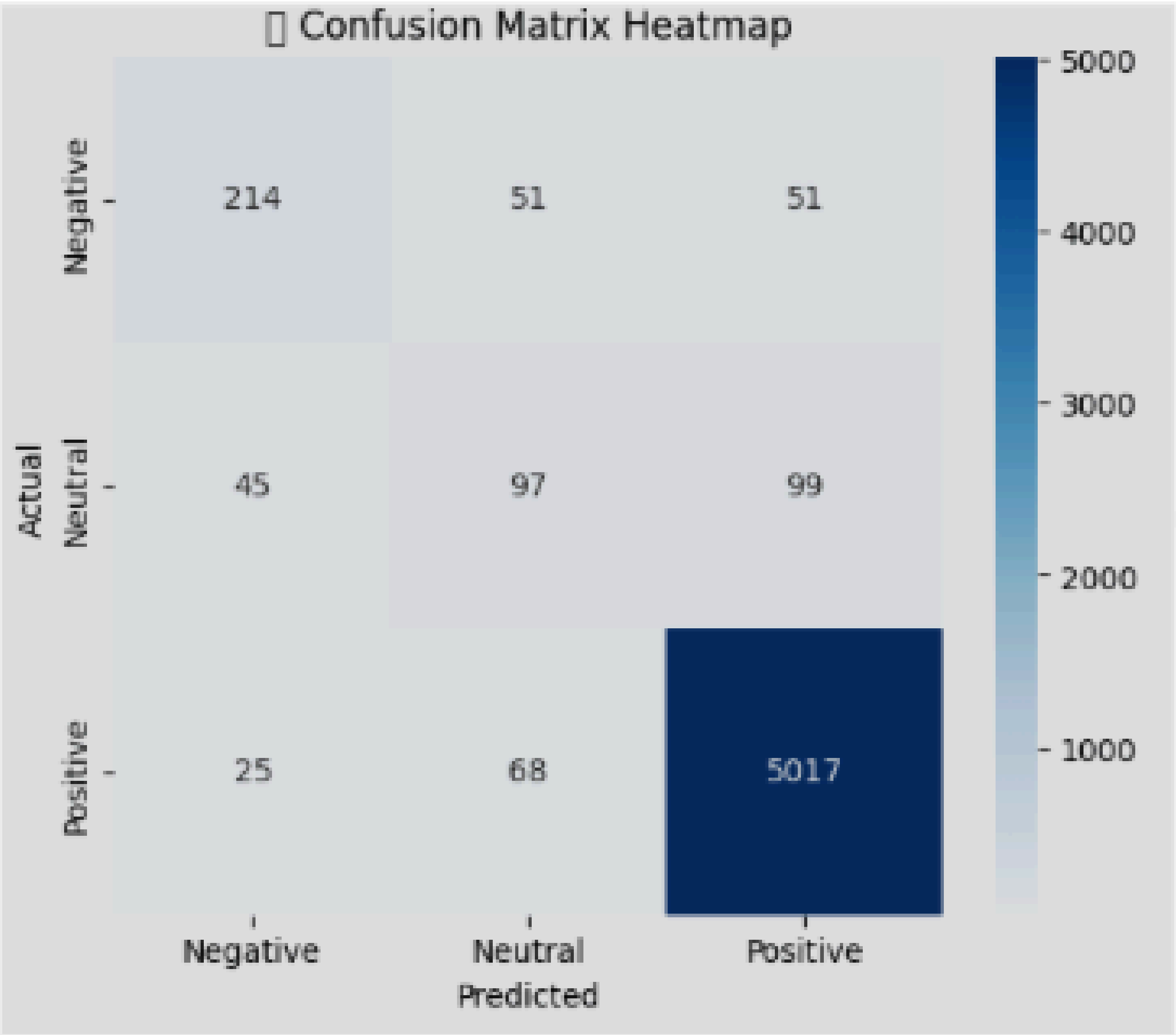


Exploratory Data Analysis

Accuracy: 94.02%

Classification Report:

	precision	recall	f1-score	support
Negative	0.75	0.68	0.71	316
Neutral	0.45	0.40	0.42	241
Positive	0.97	0.98	0.98	5110
accuracy			0.94	5667
macro avg	0.72	0.69	0.70	5667
weighted avg	0.94	0.94	0.94	5667



Product Category Clustering

Major Obstacle

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

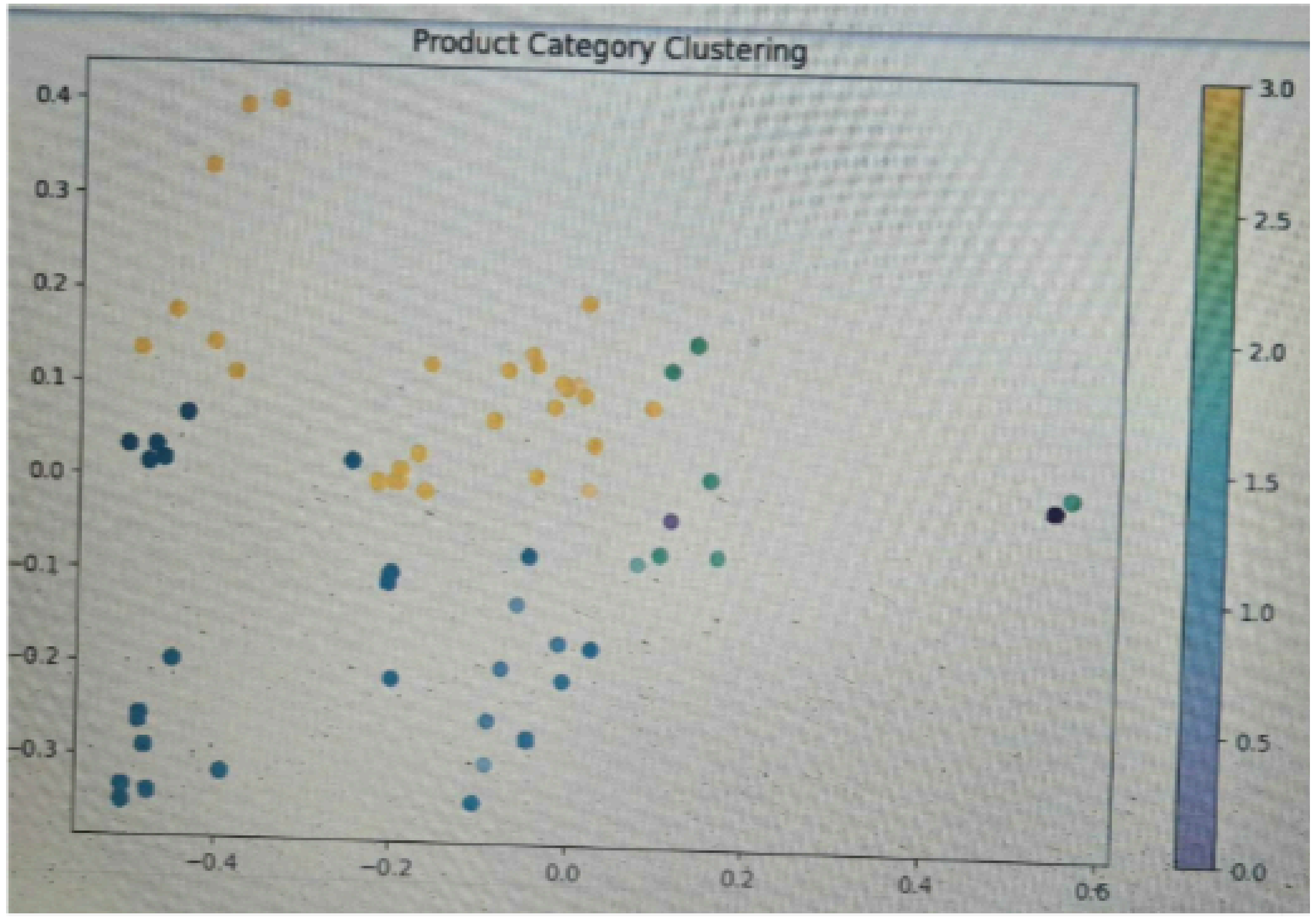
1. Technical Insights
   - Importance of thorough data preprocessing
   - Value of efficient model selection
   - Need for robust error handling

2. Process Improvements
   - Better initial planning
   - More comprehensive testing
   - Regular performance monitoring

Summary of each model

1. `distilbert-base-uncased`

- **Library**: `transformers` (by Hugging Face)
- **Purpose**: Sentiment classification of customer reviews (positive, neutral, negative).
- **Used for**: Fine-tuned for review classification using `Trainer` and `TrainingArguments`.
- **Benefit**:
  - Provided a custom sentiment model tailored to the review dataset.
  - Improved accuracy by balancing class weights during training.



2. `all-MiniLM-L6-v2`

- **Library**: `sentence-transformers`
- **Purpose**: Convert product names and categories into numerical embeddings (vectors).
- **Used for**: Clustering products based on text similarity.
- **Benefit**:
  - Enabled grouping similar products into 4 categories.
  - Helped visualize product clusters using KMeans and PCA.

3. `facebook/bart-large-cnn`

- **Library**: `transformers` (via `pipeline("summarization")`)
- **Purpose**: Summarize review texts into structured product summaries.
- **Used for**: Automatically generating short blog-style recommendations for each product category.
- **Benefit**:
  - Created readable summaries of top-rated and worst products.
  - Highlighted key differences and pros/cons using natural language generation.

thank you

any questions?

@Sahar Sheshah
sheshah45@gmail.com