saudi digital academy

Himah Digital Bootcamps - AI Bootcamp

Business Case
Automated Customer Reviews Analysis using AI

Team Members

Ahmed Alqarni

Amal Alghtani

Hanan Alnbhani

2025

Table of contents

Contents
1. introduction
2. Objectives
3. Sentiment Classification Process
4. Product Category Clustering
5. Review Summarization
6. Deployment
7. Challenges & Solutions
8. conclusion

1. Introduction

This project is focused on analysing product reviews using Natural Language Processing (NLP). The goal is to classify customer feedback into sentiment categories (*Positive*, *Negative*, *Neutral*), group similar products using clustering techniques, and generate summaries and comparisons based on the findings. The project also includes deployment to make the system accessible and interactive

Main Features :

- Sentiment analysis of reviews.
- Clustering of similar products (4 clusters).
- Summarization and comparison of products.
- Article generation for each cluster.
- Deployment of the final application

2. Objectives

The goal of this project is to analyse product reviews by:

- Classifying sentiments into Positive, Negative, or Neutral.
- Clustering products into 4 groups based on review patterns.
- Summarizing and comparing product performance.
- Generating a short article for each cluster.
- Deploying the system with an interactive interface

3. Sentiment Classification Process

Data Selection

In this task the dataset used are:

Primary Dataset (Primary Amazon Reviews)
Larger Dataset: (Amazon Reviews Dataset)

We initially used the Amazon Product Reviews dataset as our primary data source. It included a wide variety of customer reviews with star ratings, making it ideal for sentiment analysis. And for solving the unbalance problem we upload and use the larger dataset.

To analyse and classify customer opinions, we developed a sentiment classification model using product review data.

The following steps summarize our approach:

1. Choosing the Primary Dataset

We initially used the Amazon Product Reviews dataset as our primary data source. It included a wide variety of customer reviews with star ratings, making it ideal for sentiment analysis.

2. Data Cleaning and Preprocessing

We removed unnecessary symbols, duplicate entries, and empty reviews.

The focus was placed on the most relevant columns: review text and rating.

3. Sentiment Mapping

To simplify the task, we converted the star ratings into three sentiment categories:

1–2 stars → Negative

3 stars → Neutral

4–5 stars → Positive

4. Data Balancing

The original dataset was imbalanced, with positive reviews significantly outnumbering the others.
To ensure fair training, we supplemented the data using a larger dataset (Amazon Reviews Dataset) to increase the number of negative and neutral reviews. This step helped reduce model bias and improve classification accuracy.

5. Model Selection: BERT

We selected the BERT model due to its strong performance in understanding context and language.

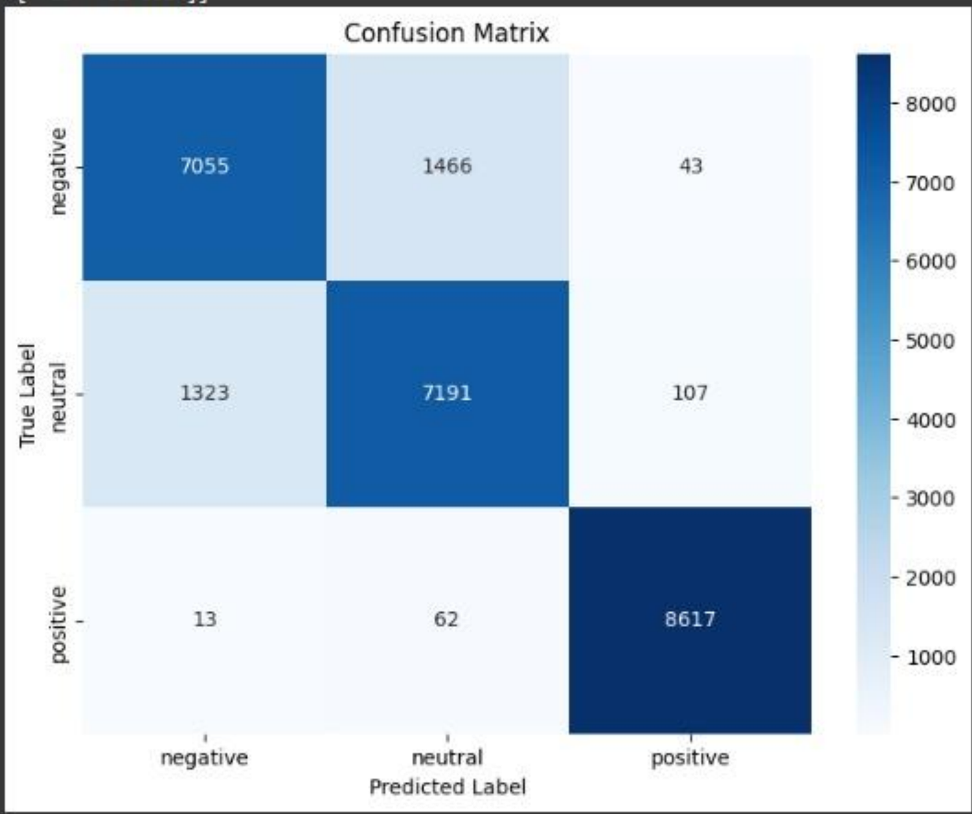
It outperformed other models, including DistilBERT, and proved highly effective for sentiment analysis tasks.

6. Training and Evaluation

The model was trained using the balanced dataset. Evaluation results, including the confusion matrix and classification metrics, are presented in the attached visuals. These results confirm the model's ability to accurately distinguish between the three sentiment types.

Model Evaluation

```
Confusion Matrix:
 [[7055 1466   43]
 [1323 7191  107]
 [  13   62 8617]]
```




```
[ ] trainer.train()
```

[16175/16175 15:37, Epoch 5/5]

Epoch	Training Loss	Validation Loss	Accuracy	Precision Negative	Precision Neutral	Precision Positive	Recall Negative	Recall Neutral	Recall Positive	F1 Negative	F1 Neutral	F1 Positive
1	0.269200	0.325848	0.864126	0.804822	0.802180	0.983479	0.810836	0.793991	0.986194	0.807818	0.798065	0.984835
2	0.229100	0.323769	0.879275	0.846135	0.810029	0.980736	0.805231	0.841318	0.989876	0.825176	0.825377	0.985285
3	0.149800	0.369678	0.883410	0.843912	0.819034	0.986462	0.818192	0.841550	0.989185	0.830853	0.830139	0.987822
4	0.102500	0.444712	0.883526	0.840782	0.824751	0.982890	0.823797	0.834126	0.991371	0.832203	0.829412	0.987113
5	0.062400	0.515882	0.883487	0.843389	0.821372	0.984116	0.819360	0.838998	0.990796	0.831201	0.830091	0.987445

[3235/16175 02:51 < 11:25, 18.88 it/s, Epoch 1.00/5]

Epoch	Training Loss	Validation Loss

[809/809 07:35]

```
TrainOutput(global_step=16175, training_loss=0.1666227218061916, metrics={'train_runtime': 937.6343, 'train_samples_per_second': 551.958, 'train_steps_per_second': 17.251, 'total_flos': 3.453639876768614e+16, 'train_loss': 0.1666227218061916, 'epoch': 5.0})
```

4. Product Category Clustering

As part of the project requirements, we were tasked with grouping products into 4 to 6 clusters. The goal of this step was to classify similar products based on how users described them in their reviews.

1. Data Selection

We used the same primary dataset (Primary Amazon Reviews) as in the sentiment classification task. In this step, we focused on two columns:

- **Categories** (to understand product type)
- **Review Text** (actual user feedback)

2. Text Embedding

To convert review text into a numerical representation suitable for clustering, we explored several text embedding models. After evaluation, we selected intfloat/e5-small-v2, a lightweight and efficient model from Hugging Face. It performed best on our data, successfully capturing semantic similarities between reviews.

3. Clustering Algorithm

We applied the **K-Means** algorithm, initially generating **5 clusters**. After analysis, we found that some clusters were semantically similar and contained overlapping content.

To improve clarity and reduce duplication, we merged similar clusters, resulting in **4 final clusters**, each representing a distinct product theme.

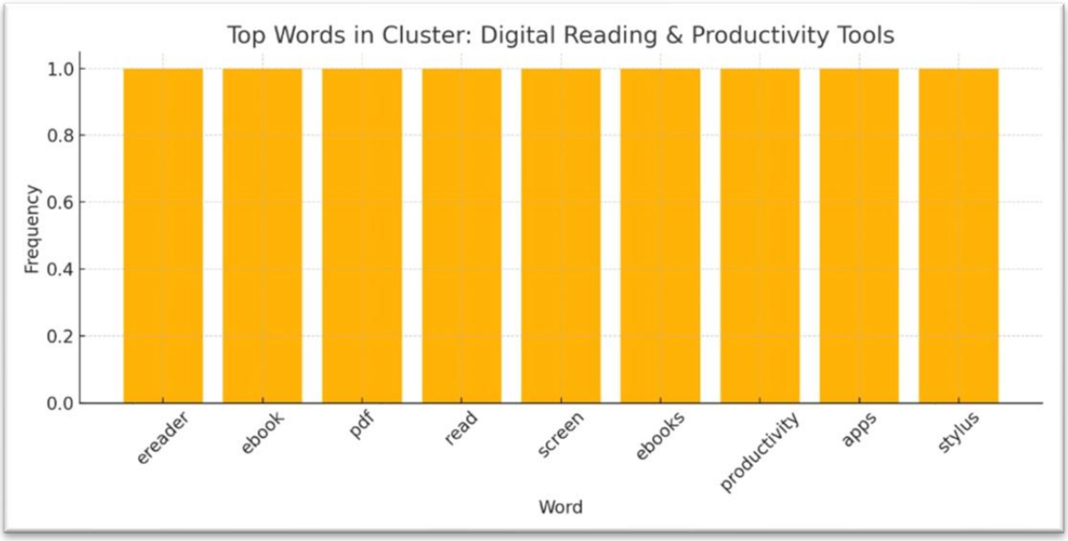
4. Defining the Final Clusters

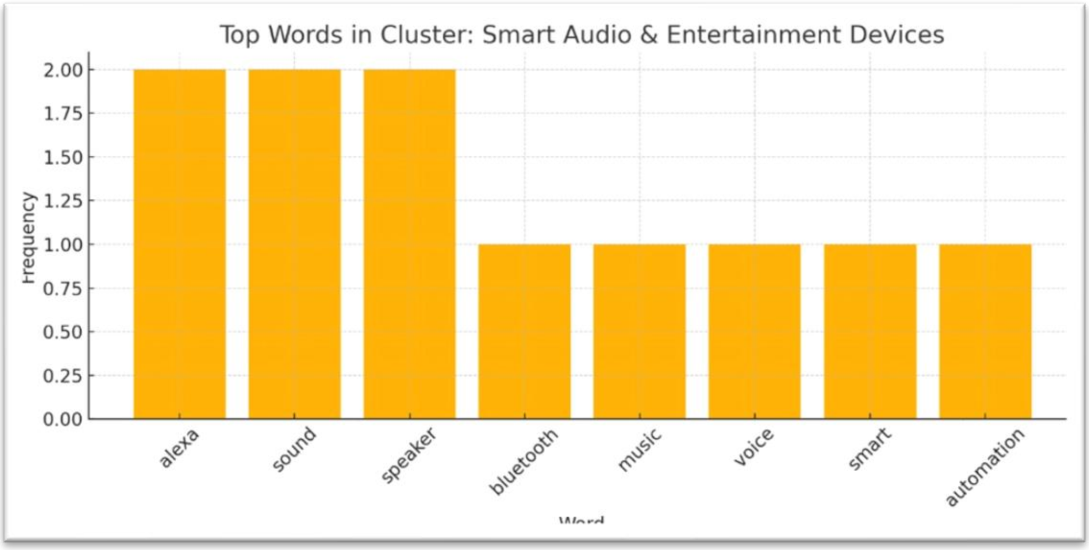
The reviews were categorized into the following product categories:

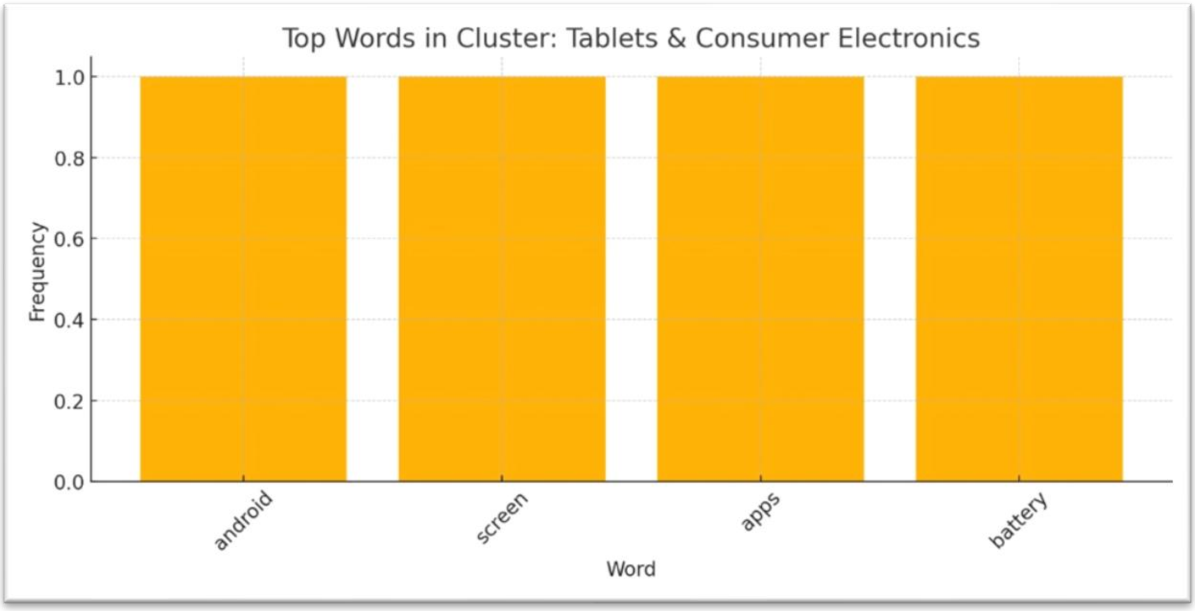
- "Digital Reading & Productivity Tools"
- "Smart Audio & Entertainment Devices"
- "Tablets & Consumer Electronics"
- "Streaming Devices & Media Playback"

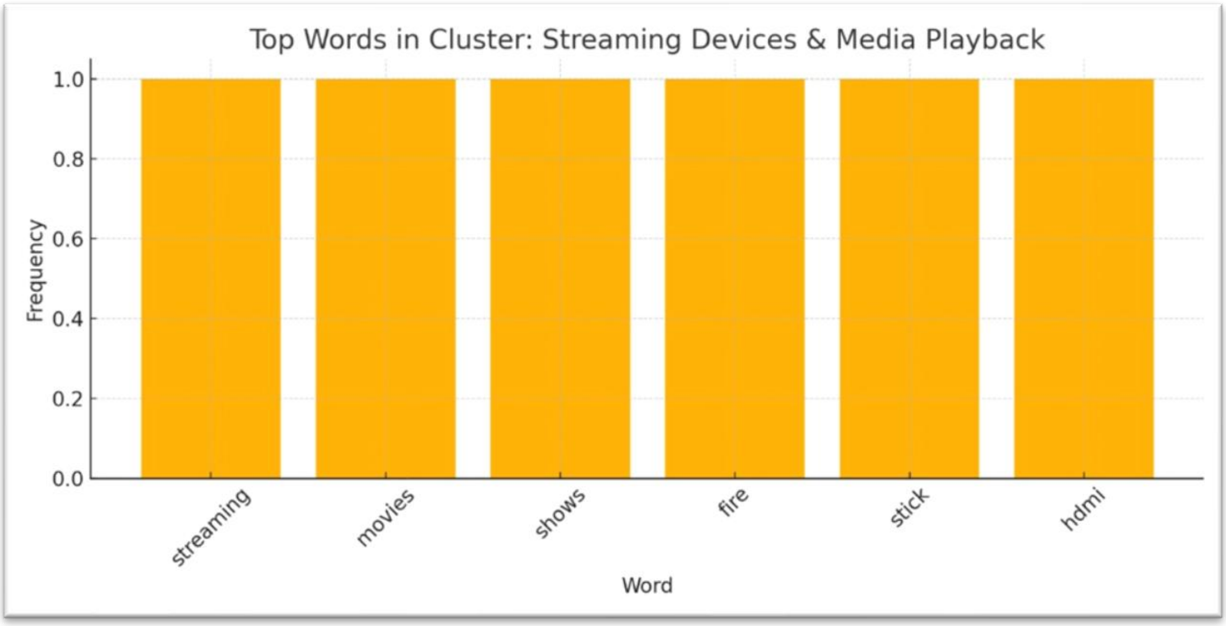
To better understand the focus of each product cluster, we extracted and visualized the most common words in the reviews.

These recurring terms helped us identify the types of products discussed within each cluster and provided insights into what users were primarily talking about. For example, some clusters included frequent references to "tablet," "ebook," or "Kindle", which clearly aligns with the 'Readers & Tablets' cluster theme.









5. Review Summarization

In the stage of our project, we used the generated clusters to organize the reviews into clear product categories. Then, we evaluated multiple text generation models — including GPT-3, T5, and BART — using a loop-based comparison. Based on the output quality, GPT-3 consistently produced the most coherent and informative summaries.

We used GPT-3 to generate short blog-style articles for each category. Each article included:

The top 3 products and their key differences.

The main user complaints about each.

The worst product in the category and why customers disliked it.

The GPT-3 model proved to be highly effective at transforming raw customer feedback into well-structured and insightful summaries

6. Deployment

To make our system interactive and user-friendly, we deployed the final model using Gradio. The interface allows users to first upload a CSV file containing customer reviews. Once uploaded, users can select a product category from the clustered data and instantly view the AI-generated summary for that category. This setup makes the tool flexible and easy to use for various datasets.

7. Challenges & Solutions

Challenges	Solutions
1. Dataset Imbalance	Use rebalancing techniques such as oversampling or under sampling.
2. Large-scale review datasets (like Amazon Reviews) require high computational resources and can slow down training and testing.	Use lightweight models like BERT to reduce resource consumption.
3. Difficulty in choosing the appropriate number of groups for clustering	Use criteria like Silhouette Score and Elbow Method to automatically select the optimal number.

8. conclusion

This project showcased the power of Natural Language Processing (NLP) in transforming raw customer reviews into structured, insightful summaries. By combining clustering techniques with advanced generative models like GPT-3, we were able to identify top-performing products, highlight user concerns, and generate meaningful recommendation articles. The system enhances user experience, saves time in manual analysis, and provides businesses with actionable insights derived directly from customer feedback. This approach lays the foundation for more intelligent review analysis tools in the future.