Group 3
RoboReviews
A Model That Summarizes Amazon Reviews

Group Members: Ben Paffrath and Sofia Stephenson

Smart Product Decisioning –
Find the Best Product in Seconds

What we do:
We quickly identify the best products in any category – and generate a tailored AI recommendation for your customer.

What makes us unique:
We don’t just list options – we deliver a clear recommendation, based on real user feedback and smart AI analysis.

The Result:
Users decide faster, trust the recommendation –
and convert more often.

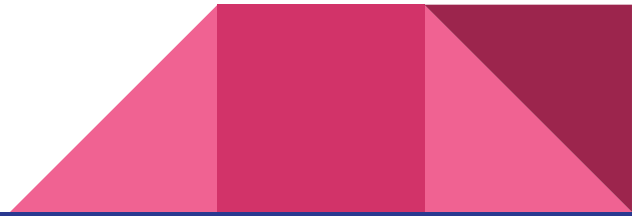
Introduction

The problem:

- Too many products
- Too little guidance
- Customers feel overwhelmed and drop off

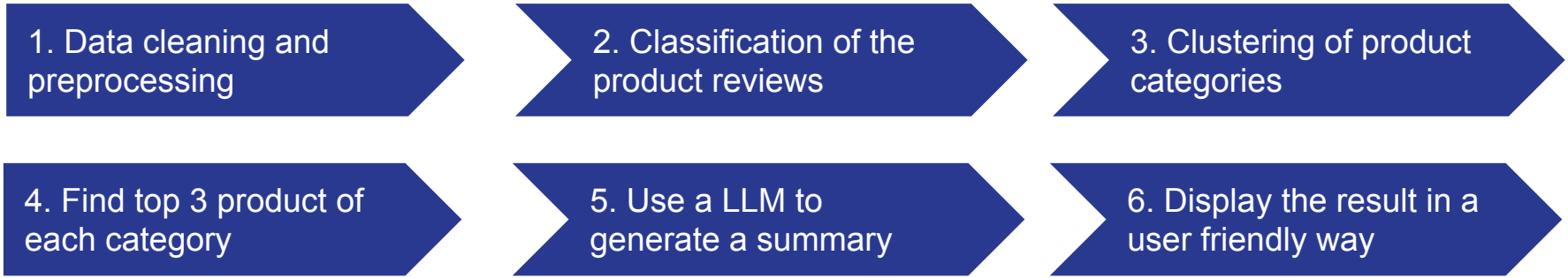
The solution:

- Analyze reviews
- Find the top products
- AI generated recommendations

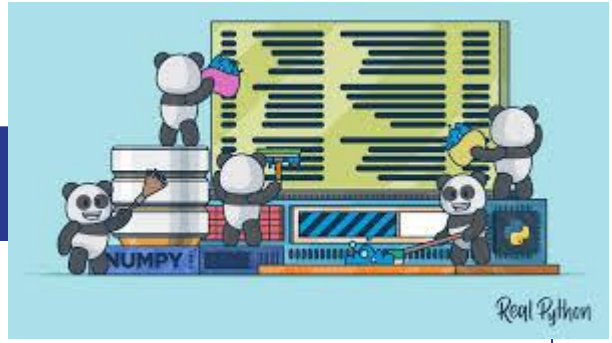


Methods / Pipeline

- We chose Dataset 1
  - Smaller and easier to handle
  - Still enough data to work with
  - Combined all three files
  - Quality not good, but manageable

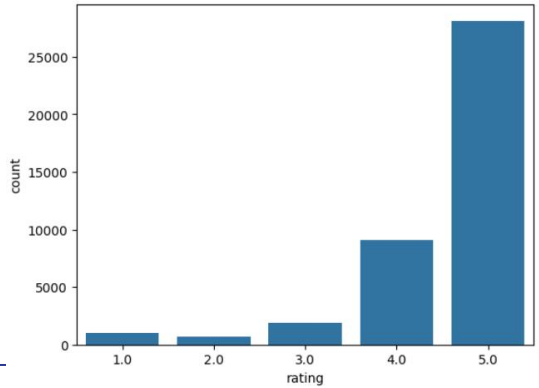


Data Cleaning / Preprocessing



Methods

- Removed duplicates & empty rows
- Check for unique ratings to ensure consistency
- Combined text columns
- Encode categorical variables (e.g., convert categories to numerical codes).
- Text preprocessing
  - Removed HTML
  - tokenized
  - Removed stop words and punctuation
  - Applied lemmatization
- TFidf-Vectorization
- Handle imbalanced data with SMOTE



Model 1

Classification

The Problem:

The reviews all had a rating of 1-5, but to be able to make a really good recommendation we don't need the neutral ratings, for example, and it makes sense to categorize the ratings into positive and negative ones.

Model 1 - XGBoost

Classifying the Reviews

The Solution:

We created a new column and divided the reviews into 3 different categories.

- Negative (1, 2)
- Neutral (3)
- Positive (4-5)

Accuracy: 93.71 %

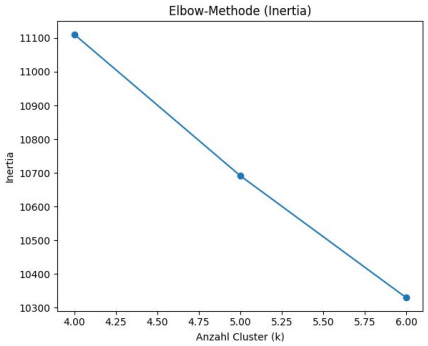
Model 2

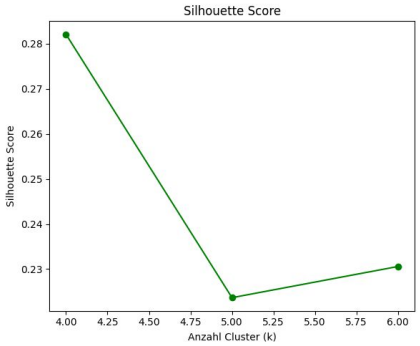
Clustering

The Problem:

There are more than 100 categories in the data set, which is too many to work with properly.

Model 2 - K-Means





The Solution:

We want to group similar product categories together.

- Embeddings
- Finding optimal clusters
- Grouped products
  - Entertainment Tablets
  - E-Reader & Office Tablets
  - Health & household accessories
  - Smart Home & Amazon devices
- Identified the top 3 products

Model 3

Prompt-Engineering

The Problem:

We want to use a small LLM to generate summaries of our products based on the reviews in the data that resembles a blog post from Wire Cutter or another professional journal for product reviews

Model 3 -
T5-small, Flan-T5- base, Flan-T5-Large

The Solution:

Prompt-engineering using multiple small models that specialise in summarising data.

Findings:
These LLMs were insufficient for this task ask they were overly influenced by the linguistic shortcomings of the cleaned and preprocessed data. They were not greatly swayed by promptly engineering.

Takeaway

Classification: XGBoost → Clustering: K-Means → Summary: Llama 3.1 8B

- If there is no data set to fine-tune, it is a huge amount of work
- Prompt engineering with small models is challenging
- Base-Models fine tuned on instructions | game changer
- Model hosting is expensive
- Split instructions into multiple tasks
  - Summarization for each product
  - Final Summary + Recommendation

Demo

https://ironhack-project-4-g3.benpaffrath.de/

Any Questions?

Group 3: Ben Paffrath and Sofia Stephenson