

Robo-Reviews

Javi Ansoleaga



Project Overview

RoboReviews is a project designed to analyze customer reviews using AI to generate a blog-style recommendation of the top products along different product categories.

Main Goals:

1. **Sentiment Analysis:** Classify reviews as positive, negative, or neutral to understand customer sentiment and use it later to identify the best products.
2. **Product Clustering:** Cluster reviews into meaningful product categories to be able to later identify the best one in each category.
3. **Review Summarization:** Use generative AI to summarize reviews and recommend the top products for each category based on the inputs of the previews models.

RoboReviews:

Turning customer reviews into... Something, eventually...

Why This Is Still Awesome:

- Despite the challenges, the groundwork is in place. Every good project starts somewhere.
- If nothing else, I've learned a whole lot during this project. Specially how *not* to build a model or two.
- Plus, useful for the project or not, the preprocessing is almost (sadly...) a project on its own.



What is your story?

Model 1: Sentiment Analysis

This one's functional! It can tell a happy customer from an angry one most of the time. It just struggles a little bit with the personality-less users writing Neutral reviews.

Model 2: Product Category Clustering

Technically, there's code that runs start to finish. Realistically, it's just a very confused collection of products that refuses to behave. Consider it a philosophical experiment into chaos.

Model 3: Generative AI for Summarization

Barely started, but the ideas are solid. If ideas could be considered finished models, then I'd have a masterpiece!

Demo (Sentiment Analysis)

404 – Proper Demo Not Found

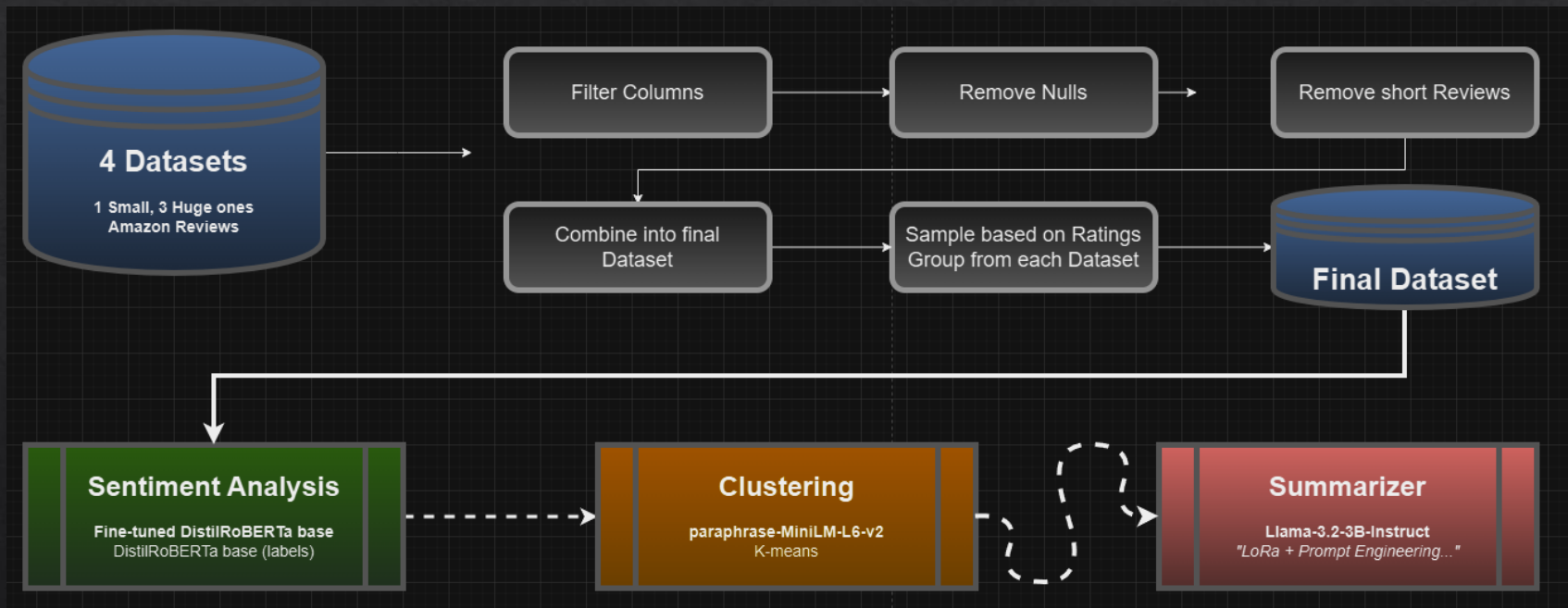
REVIEW:

Good for labeling boxes with each room, makes it easy for moving or storing but you have to be patient when using, like I said it splits/tears when pulling it off the roll.

CLASSIFICATION:

POSITIVE: (prediction score: 0.9864)

Methods



Model 1: Sentiment Analysis

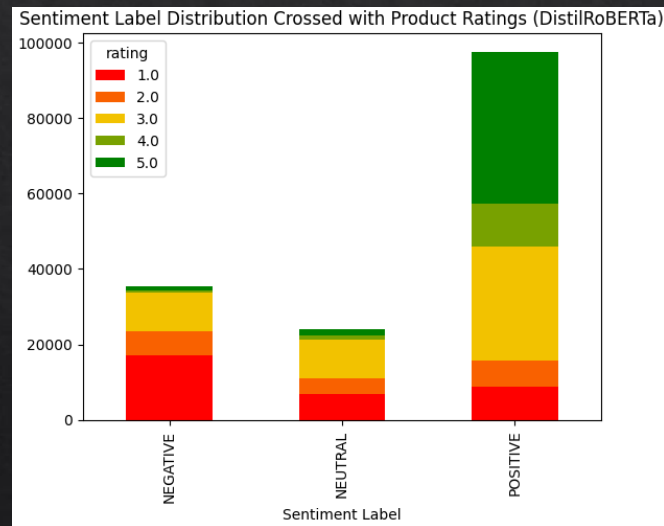
The goal was to classify customer reviews into positive, negative, or neutral categories.

Problems:

- The review ratings are not always consistent with the sentiment.
- Lack of True sentiment labels for training the model.

Approach:

- Generate labels with a simpler approach to then train and evaluate the model.
- **VADER**: Initially used, but it struggled with nuanced sentiment.
- **DistilRoBERTa**: Switched to using the base model before training to generate the labels.



Model 1 - Evaluation

- VADER +
DistilRoBERTa

Epoch	Training Loss	Validation Loss	Accuracy
0	0.652000	0.585210	0.750366
2	0.456400	0.451295	0.846159
4	0.357300	0.418835	0.863376
6	0.286000	0.424632	0.863281
8	0.248900	0.436297	0.876551

True label	Predicted label		
	NEGATIVE	NEUTRAL	POSITIVE
NEGATIVE	5666	714	118
NEUTRAL	753	3213	384
POSITIVE	481	1912	18181

- DistilRoBERTa +
Fine-tuned
DistilRoBERTa

Epoch	Training Loss	Validation Loss	Accuracy
0	0.507100	0.466671	0.821113
2	0.249100	0.232423	0.928140
4	0.167100	0.215747	0.931417
6	0.131000	0.214208	0.935364
8	0.110800	0.227629	0.934504

		Predicted label		
		NEGATIVE	NEUTRAL	POSITIVE
True label	NEGATIVE	6535	315	229
	NEUTRAL	405	4220	193
	POSITIVE	330	778	18417

Takeaway

Project Recap:

- Model 1:
Functional but not Perfect! Managed to classify reviews into sentiment categories.
- Model 2:
Technically built, practically a bit of a mess (clustering didn't quite work out as planned).
- Model 3:
Barely started, but I laid some foundations...



Takeaway: Mistakes where made...

- Got lost with which datasets to use and how and spent way too much time on data preprocessing to get the perfect (or so I believed) dataframe. Which the Project was definitely not about.
- Got also confused with what models or ML techniques to use for what and their different implementations.
- Got distracted by lots of investigation that went nowhere, learning better ways of doing things and side experiments instead of focusing on the project goals.
- **Main Lesson:** Finish a simple and working versión first, then optimize.

Learned:

- Despite running out of time for everything, I learned a lot about all the differenet models and approaches, and specially about what **NOT** to do in a project like this.

Repository: [Grimngor/RoboReviews \(github.com\)](https://github.com/Grimngor/RoboReviews)