

2023-02-23

Using AI to Automate a Personalized Klaviyo Campaign

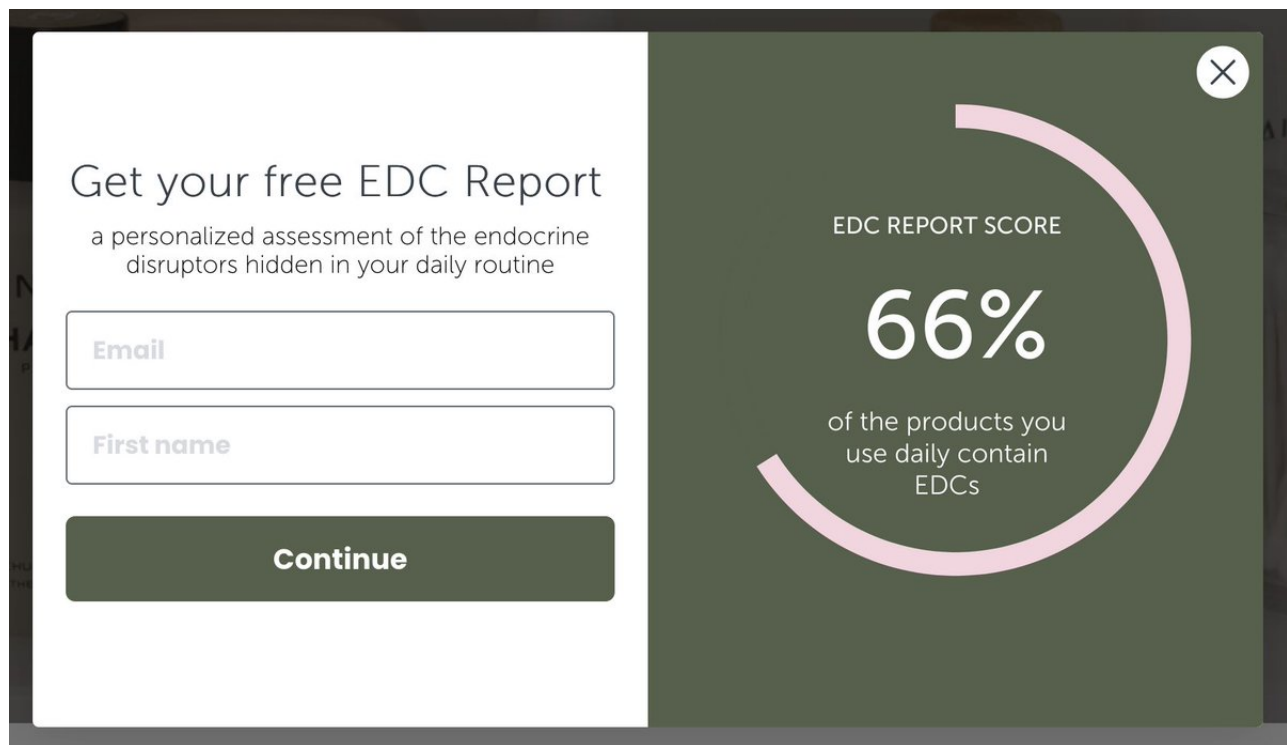
tldr; With GPT-3, Supabase, Pinecone, and Klaviyo, I built a semantic embeddings search that saved hundreds of hours of manual work for my girlfriends skincare brand ([Oliver Care](#)).

It actually worked way better than I expected - this is how I did it:

First some context. My girlfriend, [Maria](#), started Oliver Care with the goal of creating a brand that educates people on the risks of Endocrine Disrupting Chemicals (EDCs) in every day items and sells EDC-free products.

As one of her marketing campaigns, she had the idea of doing a personalized EDC report where anyone could send in their daily routines and she'd look at each item and identify which contained EDCs and make suggestions.

At first she'd get a couple a day, from [her TikTok following](#), but that quickly grew to thousands 🌍



At that point, we realized she'd have to automate it somehow. I quickly pulled all the data together, and started writing up a python script.

We had all the email addresses and their submitted routines in Klaviyo which I exported to Supabase. I got her to put down the most popular products she had already seen and her response to each and loaded as well.

COMMON PRODUCT RESPONSES	
Product	Description
Cerave	depending on where you live
Paula's choice 2% BHA	contains salicylic acid (known
Dove bar soap	contains undisclosed fragrances
Redken all soft Shampoo/conditioner	contains undisclosed fragrances
Olaplex No.3	contains Butylphenyl Methyl
Olaplex No.4	contains Hexyl Cinnamaldehyde
Olaplex No.5	contains BHT (suspected allergen)
Olaplex No 6	contains undisclosed fragrances
Olaplex Bonding Oil	contains undisclosed fragrances
Ouai Hair Oil	contains Butylphenyl Methyl
Supergoop Unseen Sunscreen	contains Avobenzone (known

I was hoping I'd be able to write a quick script to do some regex parsing and use the `pg_trgm` extension on Supabase to look up each product in a person's routine and match it to a product responses she'd filled in.

That worked exactly as expected - text search is actually super easy with Postgres.

... But text search isn't really enough. People would send in product names formatted in all types of ways. And there's multiple names for the same product. Text search isn't really well suited to that.

Next I thought, maybe GPT-3 could automate this completely.

Find a matching item for the list below

Carina Organics

Sol De Janeiro Body Butter

Inkley List vitamin C

L'Occitane

Elf Daily Hydration Moisturizer

loccitane moisturizer: matches L'Occitane

head and shoulder: no match

elf moisturizer: matches Elf Daily Hydration Moisturizer

inkley vit c: matches Inkley List vitamin C

Find a matching item for the list below

Carina Organics

Sol De Janeiro Body Butter

Inkley List vitamin C

L'Occitane

Elf Daily Hydration Moisturizer

loccitane moisturizer: matches L'Occitane

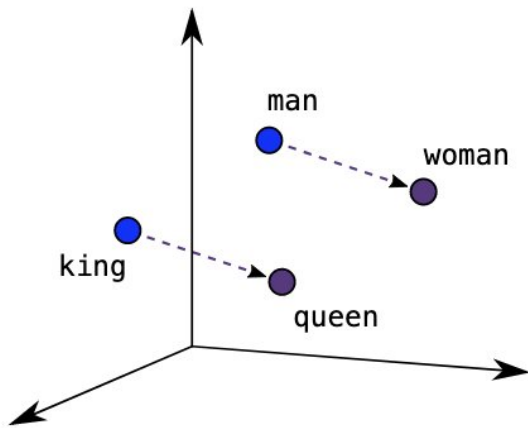
head and shoulder: no match

elf moisturizer: matches Elf Daily Hydration Moisturizer

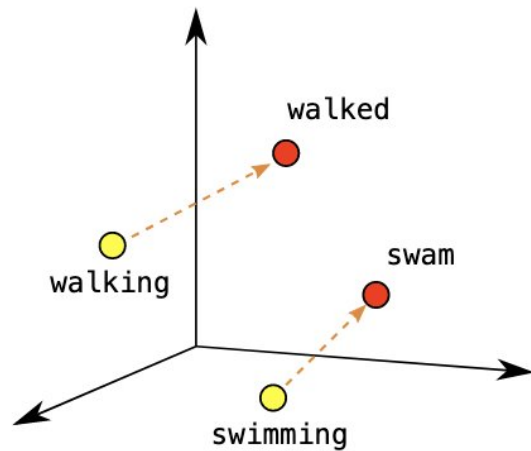
cerave moisturizer: no match

And it pretty much worked! Using prompts to find matches is not really ideal, but we want to use GPT's semantic understanding.

That's where Embeddings come in.



Male-Female



Verb Tense

Embeddings are the dense vector representation of an input encoded by the model, how the model represents a concept.

Since embeddings are just vectors, you can calculate distances between them which represent how conceptually similar they are from each other.

OpenAI exposes embeddings calculation for each of their GPT-3 models as an API - but you still need to do the vector distance calculation yourself. You can do it in code, iterating through all the potential matches and calculating the cosine distance, or you can use a vector DB.

For that I gave Pinecone a try, which is built for this exact purpose. My script read through each of the products we had responses for, called

OpenAI's embedding api and loaded it into Pinecone - with a reference to the Supabase response entry.

Once the response lookups were loaded into Pinecone, the script could query it for the most similar match for each item in someone's routine and pre-populate the response for their submission.

From there we just had to run it for the thousands of submissions, export the results as a custom property of each Klaviyo user profile and kick off a campaign to email them their results.

Just like that, 100s of human hours saved using GPT-3 and Pinecone



Lessons

People normally focus on GPT-3's generative ability to produce content, but using it for this made me realize how many use cases could take advantage of it's semantic understanding of concepts.

One thing I'd love to play around with is extracting dimensions of an embedding, ie. for "cerave moisturizer", being able to calculate the "product" (moisturizer) distance to something vs the "brand" (cerave) distance.

While you're here, if you care about [protecting your hormone health](#), [check out Oliver Care](#) :)

Reach out to me on [Twitter](#), I'd love to hear from you!