

## مقدمه

در این داکيومنت تسک هم‌تیمی فنی موری رو توضیح می‌دهیم.

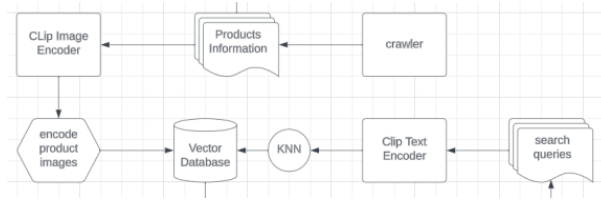
- بخش اصلی انتخاب هم‌تیمی ما همین تسک است. دلیل این تصمیم این است که معمولا در فضای فنی آدم‌ها راحت‌تر هستند در محیط واقعی ارزیابی شوند و هم‌چنین شما نیز کاملا با جنس کارهایی که ما در تیم داریم آشنا می‌شوید و تصمیم‌گیری برای این پوزیشن را راحت‌تر انجام می‌دهید.
- زمان این تسک 3 روز است. از آن‌جا که حجم این تسک به نسبت زیاد و پیچیده است، می‌توانید به این تسک با دید انجام یک پروژه جالب در فیلد خودتان نیز نگاه کنید. از سمت ما پابلیش کردن کدها و دموها در گیت‌هاب و ... خودتان مشکلی ندارد (و توصیه می‌شود:)).
- در این تسک ما هدف، کلیات و مدل‌ها را توضیح می‌دهیم؛ باقی موارد مانند ساختار کد، انتخاب استک‌های مختلف و ... به انتخاب شما است.
- این تسک تلاش شده‌است در کنار نوآورانه بودن، well known هم باشد؛ در نتیجه اگر جایی از آن برایتان جدید بود سرچ کنید و احتمالا چیزهای به‌دردبخور و جالبی پیدا خواهید کرد. :))
- در تسک رقابت وجود دارد، در نتیجه سعی کنید از جهاتی که خودتان تصمیم می‌گیرید، بهترین نسخه‌ای که با توجه به وقت و انگیزه و علاقه خودتان دارید را پیاده‌سازی کنید.

## توضیح تسک:

در این تسک می‌خواهیم یک search engine خاص پیاده‌سازی کنیم. بدنه اصلی این سرچ یک سرچ متن‌محور در تصاویر است.

## بخش پایه‌ای:

هدف اصلی این است که با استفاده از مدل انکودر clip openai یک سرچ‌انجین پیاده‌سازی و دیپلوی کنیم که با یوزر بتواند با تایپ کویری در عکس محصولات جستجو کند و نتایج را ببیند. اطلاعات محصولات به این داکيومنت ضمیمه می‌شود. (اطلاعات 10کا پروداکت ضمیمه شده‌است، اگر برای پیاده‌سازی زیاد است می‌توانید دیتای کمتری از این 10کا بردارید.)



کامپوننت‌های این سرچ‌انجین به ترتیب زیر باید باشد:

- یک دیتابیس وکتوری (پیشنهادها: qdrant, pinecone) تا داده‌ها و وکتورهای محصولات را نگه دارد.
- نیاز به یک مازول انکودر عکس و متن (مدل: clip openai) است، تا تمامی عکس‌های محصولات را انکود کند و در دیتابیس وکتوری ذخیره کند. هم‌چنین نیاز به انکودر متن نیز در زمان جستجو نیاز داریم.
- پایپلاین‌های دیتا که داده‌ها را تمیز + فرمت + انکود کند (می‌توانید جاب انکود را جدا کنید) و در دیتابیس بریزد و نگه دارد.
- یک صفحه فرانت‌اند که کویری را دریافت کند و نتایج را (عکس و تایتل محصولات) نشان دهد. (مانند صفحه سرچ e-commerce ها)، هم‌چنین فیلترها (برای مثال فیلتر کتگوری، قیمت، ...) را نیز از کاربر بگیرد و بر روی صفحه جستجو اعمال کند و نتایج را آپدیت کند؛ تلاش کنید صفحه سرچ و نتایج کاربردی و از لحاظ فرانتی مناسب باشد.
- به جای کرالر ما دیتا را مستقیماً به شما داده‌ایم.

\* برای این بخش می‌توانید "implementing text-to-image semantic search with openai clip" را جستجو کنید و پیش بروید.

\* توصیه می‌شود از استراکچر مناسبی برای کدها استفاده کنید. به عنوان مثال سرویس بک‌اند، فرانت و دیتابیس‌ها جدا شود (و در بک‌اند برای endpoint ها از fastapi یا ... استفاده شود و جاب‌ها جدا شوند و ...).

## بخش امتیازی:

بخش‌های امتیازی را بر اساس اولویت در ادامه می‌نویسیم:

## تبدیل شدن hybrid search engine

حال فرض بگیرید می‌خواهیم از یک keyword search در بخش استفاده از تایتل محصولات نیز استفاده کنیم و یک hybrid search پیاده‌سازی کنیم! فرض بگیرید اگر قرار است در یک درخواست از بک‌اند،  $m$  محصول از سمت سرچ‌انجین به دست بیاید،  $n$  تا از محصولات از سرچ بخش اول -سمنتیک سرچ پیاده‌سازی شده- به دست بیاید و  $m-n$  تا از سرچ keyword search بر روی محصولات. برای keyword search نیز توصیه می‌شود از meilisearch استفاده کنید و در حالت دیفالت ریزالت خوبی می‌گیرید (فیلترها را نیز باید هندل کنید).

- توجه کنید این بخش به این شکل پیاده می‌شود که انگار ما **کاملا** دو فائل سرچ جدا (1-سمنتیک سرچ متن به عکس، بخش پایه‌ای 2- کی‌وور سرچ ذکر شده در اینجا) داریم و در لایه گرفتن محصولات بک‌اند هر دو را مستقلا کال می‌کند و ریزالت‌ها را اجتماع می‌گیرد.

برای پیاده‌سازی این بخش، علاوه بر کامپوننت‌هایی که در بخش پایه‌ای ذکر شد. نیاز به یک کامپوننت برای گرفتن کویری از سمت یوزر و گرفتن داده‌ها از سمت دو سرچ‌انجین پیاده‌سازی شده و درست کردن نتیجه نهایی و دادن به فرانت است، سعی کنید این بخش را قابل کانفیگ پیاده‌سازی کنید.

## به سوی Intelligence و فراتر از آن

فرض بگیرید می‌خواهیم بفهمیم:

- در hybrid search چند درصد به semantic image و چند درصد به keyword search اهمیت بدهیم. این به این معنی است که با توجه به کویری یوزر چقدر باید در عکس‌ها بگیریم و چقدر در متن‌ها. از آن‌جا که این تصمیم کاملا به کویری یوزر بستگی دارد، یک راه برای فهمیدن این مساله این است که از یک مدل NLP-based استفاده کنیم، از آن‌جا که نمی‌خواهیم یک مدل NLP را پیاده‌سازی کنیم از LLMهای third-party می‌توانیم استفاده کنیم. از openai gpt یا replicate.com می‌توانید برای این کار استفاده کنید. مدل استفاده می‌تواند به این شکل باشد که با پاس دادن کویری از llm بپرسیم که از هرکدام از این سرچ‌انجین‌ها چند درصد استفاده کنیم.
- با استفاده از function tools های gpt و وصل کردن فیچر فیلتر/رتریوال می‌توانیم یک سیستم بر روی سرچی که پیاده‌سازی کرده‌ایم ایجاد کنیم! به این شکل که با یوزر صحبت کند و در هنگامی که یوزر کاملا نیاز خود را توضیح داد سرچ انجام شود و نتایج به یوزر ارایه و توضیح داده شوند.

## نحوه تحویل تسک

خروجی این تسک:

- 1- یک پروژه در گیت‌هاب (کدها باید قابل دیپلوی کردن باشند)
- 2- یک فیلم دمو (دموی خود محصول، فیچرهای پیاده‌سازی شده، استراکچر کلی کد و استک استفاده شده؛ حدودا 20 دقیقه)

3- در صورت امکان دیپلومنت بر روی یک محیط url نهایی (امتیازی) (می‌توانید از هم‌روش برای این کار استفاده کنید و 100 هزار تومان اعتبار اولیه می‌دهد)، و در غیر صورت امکان دیپلوی نحوه دیپلویمنت (داکر کامپوز، سوارم، میکروسرویس یا ...)