فاز اول پروژه

طراحی سیستم‌های یادگیری ماشین

دکتر فاطمه سیدصالحی

دانشگاه صنعتی شریف

ددلاین: ۹ دی ۱۴۰۴

مقدمه و هدف کلی پروژه

هدف این پروژه، آشنایی عملی شما با چرخهٔ کامل توسعهٔ یک سیستم یادگیری ماشین، در مقیاس واقعی است.

در فاز اول، تمرکز اصلی روی ایجاد یک پایپ‌لاین داده‌ایِ قابل بازتولید، استخراج بازنمایی‌های (embeddings) اولیه از داده‌های متنی و تصویری، و توسعهٔ مدل‌های پایهٔ RAG است.

خروجی این فاز شامل دادهٔ پاک‌سازی شده، پایپ‌لاین استخراج ویژگی قابل نسخه‌بندی (config)، ذخیره‌سازی embedding ها در یک Vector DB و ایجاد سه مدل بازیابی پایه (Image-Only, Text-only, Multimodal) خواهد بود.

در این پروژه، دادهٔ محصولات و فروشگاه‌ها از وب‌سایت «ترب» به شما داده می‌شود.

تعریف مسئله

در این پروژه، هدف ساخت یک دستیار خرید هوشمند است که بتواند بر اساس نوع ورودی کاربر، محصولات مناسب را در میان داده‌های ترب بازیابی کند. بنابراین مسئلهٔ اصلی ساخت یک Product Retrieval RAG System است.
مدل باید بتواند کوئری کاربر را به فضای Embedding مناسب نگاشت کرده، و نزدیک‌ترین محصولات را بازیابی و آن‌ها را Rank کند.

Observations on Dataset

داده‌های مورد نیاز از سایت «ترب» در این لینک برای شما قرار گرفته‌اند.

در ادامه توضیحاتی راجع‌به داده‌ها و فرمت آن می‌دهیم، تا درک و شهود بهتری از آن داشته باشید.

دقت کنید که این فایل‌ها در دیتابیس به صورت رابطه‌ای (Relational) بوده‌اند، و هم می‌توانید به همین فرمت با آن‌ها کار کرده (مثلا با استفاده از SQLite یا Postgresql)، یا یک فایل unify شده بسازید و همه را در صورت راحتی و نیاز، به یکدیگر لینک کنید. (مثلا concatenate یا هر کار دلخواه دیگر)

در نهایت باید اطمینان حاصل کنید که در این پروژه و فاز بعد، داده‌های با کیفیتی دارید تا دقت مدل RAG شما، قابل قبول باشد.

توجه کنید که در این دیتاست، دو سطح محصول وجود دارد:

- Base Product که نمایندهٔ یک محصول اصلی است (مثلا «iPhone 17 256 GB»)
- Member Product در واقع همان محصول است ولی با قیمت/فروشگاه متفاوت

و همینطور ستون *random_key* در Members به ستون *base_random_key* در Base Products وصل است و ساختار اصلی Retrieval نیز می‌تواند بر همین مبنا شکل بگیرد.

جدول جستجوها (Searches)

این جدول لاگ هر صفحه از نتایج جستجو را در خود ذخیره می‌کند.

نام ستون	توضیحات
id	**شناسه یکتای هر صفحه از نتایج جستجو.** برای اتصال لاگ‌های دیگر (مثل بازدید و کلیک) به این جستجو استفاده می‌شود.
uid	**شناسه یکتای یک جستجو (شامل تمام صفحات).** این شناسه برای تمام صفحات یک جستجوی خاص، یکسان است. (برابر id صفحه صفر)
query	عبارت جستجو که توسط کاربر وارد شده است.
page	شماره صفحه در نتایج جستجو (صفحه اول از اندیس 0 شروع می‌شود).
timestamp	زمان دقیق ثبت لاگ جستجو در دیتابیس به وقت UTC.
session_id	شناسه نشست (session) کاربر

result_base_product_rks	لیستی از کدهای (base_random_key) محصولات پایه‌ای که در نتایج به کاربر نمایش داده شده‌اند.
category_id	شناسه دسته‌بندی که کاربر جستجوی خود را به آن محدود کرده است. مقدار 0 به معنی عدم انتخاب دسته‌بندی است.
category_brand_boosts	لیستی از دسته‌بندی‌ها و برندهایی که در این جستجو امتیاز بالاتری (boost) در رتبه‌بندی گرفته‌اند.

جدول مشاهدهٔ بیس (Base Views)

این جدول هر بار که کاربر بر روی یکی از بیس‌های موجود در نتایج جستجو کلیک می‌کند یک لاگ ثبت می‌کند.

نام ستون	توضیحات
id	شناسه یکتای هر رویداد بازدید از صفحه محصول پایه.
search_id	شناسه صفحه‌ی جستجویی که این بازدید از آنجا آمده است (متصل به id در search).

base_product_rk	کلید رندوم (random key) محصول پایه‌ای که مشاهده شده است.
timestamp	زمان دقیق ثبت لاگ بازدید در دیتابیس به وقت UTC.

جدول کلیک نهایی (Final Click)

این جدول اطلاعات مربوط به کلیک‌های نهایی کاربر روی محصولات فروشگاه‌ها را ثبت می‌کند.

نام ستون	توضیحات
id	شناسه یکتا برای هر رویداد کلیک.
base_view_id	شناسه‌ای که این کلیک را به یک بازدید از صفحه محصول (base_view) متصل می‌کند.
shop_id	شناسه فروشگاهی که آیتم کلیک‌شده به آن تعلق دارد.
timestamp	زمان دقیق وقوع کلیک به وقت UTC.

جدول محصولات پایه (Base Products)

این جدول اطلاعات مربوط به هر محصول را ذخیره می‌کند.

نام ستون	توضیحات
random_key	کلید شناسایی یکتای محصول پایه
persian_name	نام فارسی محصول
english_name	نام انگلیسی محصول
category_id	شناسه دسته‌بندی محصول
brand_id	شناسه برند محصول
extra_features	ویژگی‌های اضافی محصول (JSON)
image_url	آدرس تصویر محصول

members	لیستی از random_key های مربوط به محصولات فروشگاهی (Members) که این محصول پایه را پوشش می‌دهند.

جدول محصولات فروشگاه‌ها (Members)

این جدول، هر محصول در هر فروشگاه را، به محصولات پایه (جدول قبل) لینک می‌کند.

نام ستون	توضیحات
random_key	شناسه یکتای محصول در یک فروشگاه
base_random_key	کلیدی که این محصول را به محصول پایه متصل می‌کند
shop_id	شناسه فروشگاه
price	قیمت محصول در این فروشگاه

جدول فروشگاه‌ها (Shops)

این جدول، اطلاعات هر فروشگاه موجود در ترب را ذخیره می‌کند.

نام ستون	توضیحات
id	شناسه فروشگاه
city_id	شناسه شهر محل فروشگاه
score	امتیاز فروشگاه در ترب که از روی نتیجه پیگیری سفارش فروشگاه در ترب ساخته شده و عددی بین صفر تا پنج است.
has_warranty	آیا فروشگاه ضمانت ترب دارد

جدول دسته‌بندی‌ها (Categories)

این جدول، دسته‌بندی‌های کلی محصولات را ذخیره می‌کند.

نام ستون	توضیحات

id	شناسه دسته‌بندی
title	عنوان دسته‌بندی
parent_id	شناسه دسته بندی پدر. دسته‌بندی‌ها ساختار سلسله مراتبی دارند. که هر دسته، یک پدر دارد و ممکن است چندین فرزند داشته باشد. اگر پدری نداشته باشد، این فیلد مقدار -۱ دارد.

جدول برندها (Brands)

نام ستون	توضیحات
id	شناسه برند.
title	عنوان برند

جدول شهرها (Cities)

نام ستون	توضیحات
id	شناسه شهر
name	عنوان شهر

Exploratory Data Analysis (EDA)

پس از تسلط یافتن بر روی داده‌ها، مهم‌ترین کار این است که بدانیم کیفیت داده‌ها در چه سطحی قرار دارد. در این بخش شما باید نگاه دقیقی به داده‌ها بیندازید:

بررسی کنید چه ویژگی‌ها/ستون‌های مهمی وجود دارد، چه فیلدهایی به طرز محسوسی ناقص هستند، چه الگوهایی در قیمت‌ها، دسته‌بندی‌ها یا توصیف محصولات دیده می‌شود، و مشکلات رایجی مثل مقدارهای غیرعادی، تکراری بودن رکوردها یا ناهماهنگی در فرمت‌ها را پیدا کند.
EDA قرار است به ما کمک کند تصمیم بگیریم چه بخش‌هایی از داده نیاز به پاک‌سازی دارد و چه ویژگی‌هایی را باید اصلاح یا تقویت کنیم. تولید یک گزارش سادهٔ آماری و بصری در این مرحله بخش مهم کار است.

مراحل پیشنهادی EDA:

1. بررسی تعداد رکوردهای یکتا برای ستون‌های کلیدی (مثلا *random_key* یا *shop_id* یا ...)
2. تحلیل توزیع قیمت‌ها و شناسایی مقادیر غیرعادی (مثلا قیمت ۰ یا قیمت بیش از اندازه بالا)
3. بررسی فراوانی دسته‌بندی‌ها و برندها و تشخیص Class Imbalance
4. تحلیل Missingness در فیلدهای کلیدی (مثلا *image_url* یا *extra_features* یا ...)
5. بررسی consistency داده‌ها

مراحل بالا صرفا پیشنهاد است و پایپ‌لاین می‌تواند به دلخواه شما تعیین شود.

تاثیر نتیجهٔ این بخش، در مراحل بعد چشمگیر خواهد بود.

مهم: در صورت سنگین بودن کار با دیتا ها (حدود یک میلیون رکورد) و کمبود منابع برای ادامهٔ فرایند پروژه (مثل تمام شدن gpu از سمت Colab یا Kaggle یا ضعف سیستم شخصی) اجازه دارید در بخشی که نیاز به embed کردن داده‌هایتان دارید، کسری از آن را انتخاب کنید و امبدینگ بگیرید. اما انتخاب این زیرمجموعهٔ دادگان، باید منطقی، قابل توجیه و قابل دفاع، بر اساس نتایج این بخش باشد. زیرا هر subset رندومی ممکن است کیفیت سیستم و خروجی شما را به طرز محسوسی کاهش دهد.

ابزارهای پیشنهادی:

- **pandas**
- **matplotlib**
- **great-expectations**
- **pandas-profiling**

Data Cleaning & Preprocessing

در این بخش، باید با استفاده از نتیجه‌گیری‌هایی که در قسمت قبل کرده‌اید، داده‌های خام را به داده‌ای تبدیل کنید که برای مدل‌سازی و آموزش مناسب باشد.

این کار می‌تواند شامل یکسان‌سازی ساختار داده‌ها، حذف موارد تکراری، تصحیح فرمت‌ها، پر کردن یا حذف مقادیر ناموجود و استانداردسازی مقادیر مختلف باشد.
برای مثال، ممکن است قیمت‌ها در قالب‌های مختلف ذخیره شده باشند (مثلا یکی «2,000,000 تومان» و دیگری «۲ میلیون تومان» باشد) یا توضیحات محصول نیاز به تمیزسازی (نرمال‌سازی) متنی داشته باشد. هدف این است که در نهایت یک دادهٔ تمیز و منظم ساخته شود تا این پردازش‌ها همیشه بتوانند اجرا شوند (Reproducibility) و خروجی‌تان، قابل اتکا برای مراحل بعد باشد.

ابزارهای پیشنهادی:

- Pandas / Pyarrow
- Scikit-learn (ColumnTransformer, Pipeline)
- Great_expectations

Feature Engineering

مدل‌ها و سیستم‌های یادگیری ماشین عمدتاً زمانی می‌توانند عملکرد خوبی داشته باشند که ویژگی‌های معنادار و مناسب در اختیارشان قرار بگیرد. این مرحله معمولاً بیشترین اثر را روی عملکرد مدل دارد.

در این مرحله شما تلاش می‌کنید ویژگی‌های عددی قابل استفاده برای مدل را از داده‌ها استخراج کنید.

به‌دلیل اینکه داده‌ها هم شامل عکس، و هم شامل متن هستند، باید بتوانید برای هر مدالیته، مدل Encoder مناسب را پیدا کنید (از مدل‌های معروف استفاده کنید، مثلا BERT برای متن، و CLIP برای تصویر)

سپس آن‌ها را در یک Vector DB ذخیره کنید تا برای استفاده در مرحلهٔ بازیاب (Retrieval) جستجو راحت و میّسر باشد.

نکتهٔ مهم: این ویژگی‌ها باید در زمان آموزش و همچنین در زمان سرویس‌دهی (serving) همیشه یکسان تولید شوند؛ بنابراین لازم است pipeline استخراج ویژگی‌ها deterministic و قابل بازتولید طراحی شود. هر تغییری در مدل، tokenizer یا resize کردن تصویر (در صورت صلاحدید) باید نسخه‌بندی شود.

Model Selection

در این مرحله می‌خواهیم با بهترین بازنمایی (Embedding) های به‌دست آمده در مرحلهٔ قبل (برای عکس و متن)، سه سیستم RAG طراحی کنیم:

Unimodal Retrieval (Text Only)

در این سیستم، ورودی کاربر فقط یک کوئری متنی است. معمولاً مراحل این‌گونه است که:

- کوئری را encode می‌کنید (مثلا با BERT یا هر مدل خوب دلخواه)
- نزدیک‌ترین embedding های متن محصولات را بازیابی می‌کنید
- نتایج را رتبه‌بندی می‌کنید (top_k)

هدف این مدل، ایجاد یک Baseline ساده است.

Unimodal Retrieval (Image Only)

در این سیستم، کوئری تصویر است (عکس محصول)
پایپ‌لاین پیاده‌سازی می‌تواند به این شکل باشد که:

- ابتدا تصویر encode شود (مثلا با CLIP یا هر مدل دلخواه)
- بازیابی کردن embedding های تصویری مشابه
- رتبه‌بندی و نمایش نتایج

این مدل به شما کمک می‌کند که رفتار مدالیتهٔ تصویر را جداگانه بسنجید (مستقل از متن)

Bonus: Multimodal Retrieval (Text+Image)

در این مدل ترکیبی، ورودی ترکیبی از هردو مدالیته است (متن و تصویر باهم)

باید بتوانید:

- Embedding دو مدالیته را ادغام کنید (پیشنهاد: projection head, attention, concatenation یا ...)
- سپس بازیابی (retrieval) را روی آن فضای مشترک انجام دهید

این مرحله امتیازی است اما پیاده‌سازی آن، کار شما را در فاز دوم بسیار راحت‌تر می‌کند (و سیستم شما بهبود قابل توجهی خواهد داشت)

ابزارهای پیشنهادی:

- PyTorch, TensorFlow, HuggingFace
- Weights & Biases / MLFlow: Monitor experiments or models

Hyperparameter Tuning

هایپرپارامترها نقش مهمی در کیفیت مدل دارند و معمولاً انتخاب مقادیر مناسب آن‌ها به صورت دستی دشوار است. در این مرحله، شما با ابزارهای مربوطه آشنا می‌شوید تا بتوانید یک فرایند جستجوی سیستماتیک برای یافتن بهترین مقادیر طراحی کنید.

هدف این است که بتوانید درک کنید چه تنظیماتی در مدل تأثیرگذار هستند، چگونه باید فضای جستجو را تعیین کرد، و چگونه می‌توان بهترین تنظیمات را یافت.

خروجی این مرحله شامل:

- تعریف Search Space
- نمودار یا گزارش Convergence
- بهترین پارامترها
- نسخهٔ مدل منتخب

ابزارهای پیشنهادی:

- **Optuna: LINK**
- **Ray Tune: LINK**

Evaluation

در پایان، مدل RAG شما ارزیابی می‌شود. در این مرحله باید معیارهای سنجش مناسب انتخاب کنید و سپس با توجه به آن، کیفیت خروجی مدل را بسنجید و بتوانید روی خطاهای به‌دست آمده، تحلیل مناسبی داشته‌باشید.

بخش قابل توجهی از این تحلیل به‌صورت نمونه‌ای و کیفی است، مثلا می‌توانید چند سوال مطرح کنید، و سپس جواب آن را بررسی کنید.

چند نمونه: «چرا embedding متنی محصول X همیشه اشتباه بازیابی می‌شود؟» یا «در تصاویر، چه ویژگی/نویز خاصی باعث شده نتایج خوبی مشاهده نکنیم؟» یا «چه الگوهای غلطی در بازیابی بیشتر از بقیه تکرار شده‌اند؟»

بنابراین مشخص می‌شود مدل در چه بخش‌هایی موفق بوده و در چه بخش‌هایی ضعف دارد.
این ارزیابی کمک می‌کند قبل از Deployment (فاز دوم) به‌صورت عملی بفهمیم آیا مدل، ارزش استفاده دارد یا خیر!

معیارهای کاملاً پیشنهادی (بسته به مسئله و نیاز استفاده کنید):

- NDCG@k
- Recall@k
- MRR
- Precision

اصول مهندسی داده و قابلیت بازتولید

در تمام مراحل بالا، تأکید فاز اول روی ایجاد سیستمی قابل تکرار، قابل تنظیم (Configurable) و قابل گسترش (Scalability) است. شما باید بتوانید با استفاده از فایل‌های کانفیگ/تنظیمات (YAML)، نسخه‌بندی داده‌ها و مدل‌ها، ثبت محیط اجرایی و رعایت اصول مهندسی نرم‌افزار، سیستم را طوری طراحی کنید که در مقیاس بزرگ بتوان از آن استفاده کرد.

مراحل پیشنهادی:

۱. ثبت پیوستهٔ config ها: همهٔ پارامترها و مسیرها در فایل‌های yaml نگهداری شوند.

۲. لاک کردن محیط (workspace) به‌وسیلهٔ ساختن requirements یا lockfile (یا هر کار مشابه)

۳. ثبت seed ها برای reproducibility

۴. نسخه‌بندی دیتا و مدل: ارتباط بین نسخهٔ داده و نتیجهٔ مدل حفظ شود (اصطلاحاً data provenance)

۵. نوشتن چند تست ساده (عمدتاً برای فازهای بعد کارآمد است)

تا بتوان در فاز بعد آن را مستقر کرد.

چشم‌انداز فاز دوم (و آخر!)

در فاز اول توانستید دیتای تمیز، Embedding های پایه، ذخیرهٔ آن در Vector DB و سیستم retrieval اولیه را بسازید.
در فاز بعد، قرار بر این است که این سیستم را به سطح «محصول قابل استفاده» برسانید.

کارهای «احتمالی» که در فاز دوم انجام خواهید داد:

- ساخت یک سرویس کامل RAG (به همراه بخش Generation)
- استقرار Encoder ها و retrieval در محیط Production
- مانیتورینگ اتوماتیک کیفیت مدل و خطاهای مربوطه
- ساخت یک Agent ساده که بتواند بر اساس چند Skill، تصمیم‌گیری کند
- طراحی Evaluator خودکار که بتواند بر کیفیت پاسخ‌ها نظارت داشته‌باشد