

# نقشه گنج دانشمند داده: مسیری که استخدام شما در ۲۰۲۶ را تضمین می‌کند

دیگر نوشتن مدل در نوت‌بوک کافی نیست. برای ساختن آینده، باید ابزارهای جدیدی را یاد بگیرید.



دریافت داده  
(Data Ingestion)

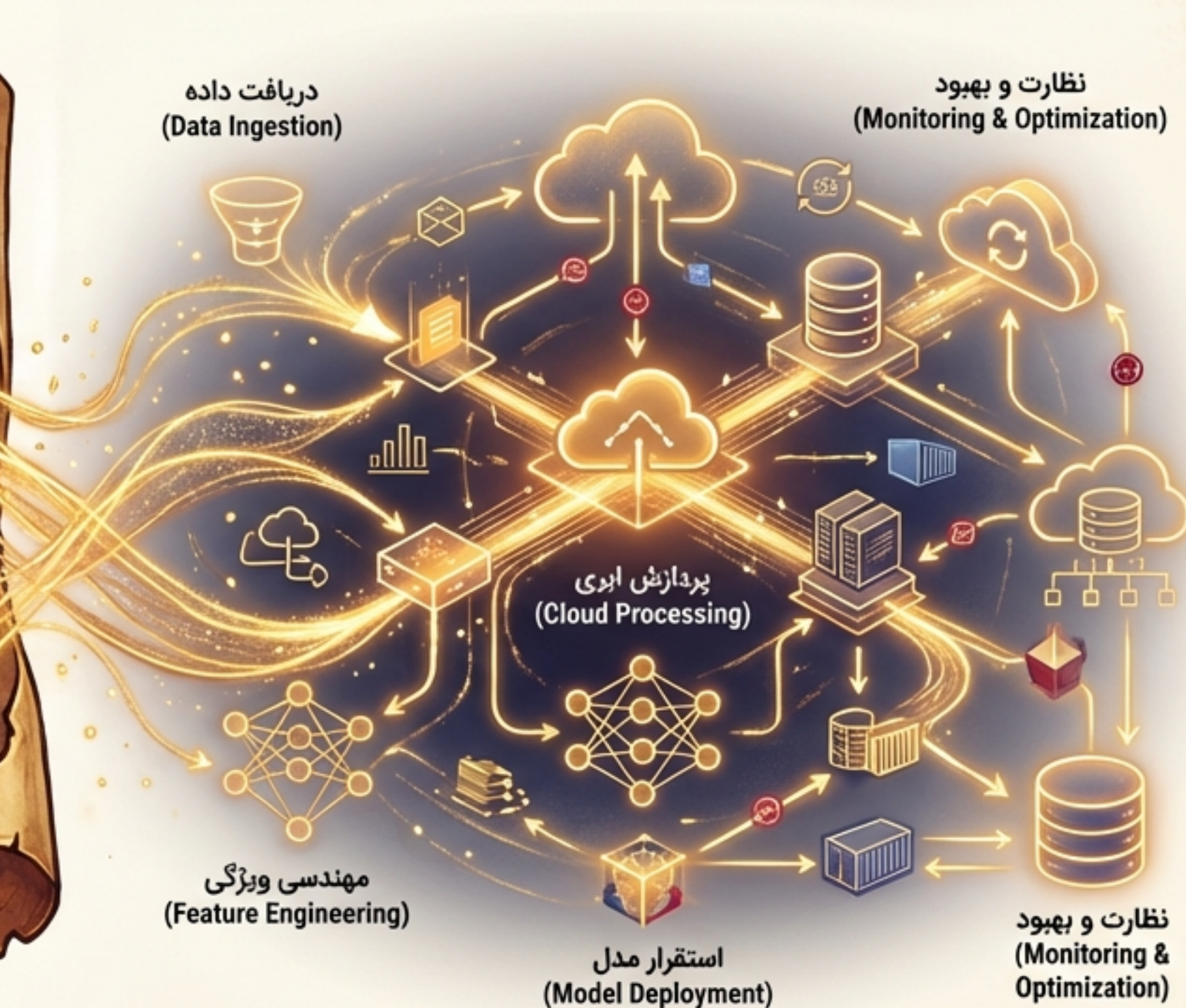
نظارت و بهبود  
(Monitoring & Optimization)

پردازش ابری  
(Cloud Processing)

مهندسی ویژگی  
(Feature Engineering)

استقرار مدل  
(Model Deployment)

نظارت و بهبود  
(Monitoring & Optimization)



# دوران نوت‌بوک‌ها به سر رسیده است

تا همین چند وقت پیش، اگر یک مدل یادگیری ماشین خفن توی ژوپیتر نوت‌بوک می‌نوشتی و به دقت (accuracy) بالایی می‌رسیدی، کار تموم بود.

اما این دنیا عوض شده.

امروز، ارزشی که خلق می‌کنی به این بستگی داره که آیا مدل تو می‌تونه از کامپیوتر شخصیات خارج بشه و در دنیای واقعی، برای کاربران واقعی کار کنه یا نه.

این یعنی باید از یک تحلیلگر محلی، به یک معمار پروداکشن تبدیل بشی.



# ذهنیت پروداکشن: یک سفر هیجان‌انگیز (و کمی ترسناک!)



ورود به دنیای پروداکشن برای خیلی از دانشمندان داده ترسناکه، چون کلی مهندسی نرم‌افزار قاطی ماجرا می‌شه. من هم دقیقاً همین حس رو داشتم. اما امروز می‌خوام یک نقشه راه صادقانه و بی‌تعارف بهت بدم. مسیری که تو رو از اصول اولیه تا استقرار کامل مدل روی کلاود می‌رسونه و مطمئن می‌شه کاری که انجام می‌دی، واقعاً ارزش تجاری ایجاد می‌کنه.

# تجهیزات سفر: ابزارهای بنیادی که باید در کوله‌پشتی‌تان داشته باشید



## پایتون

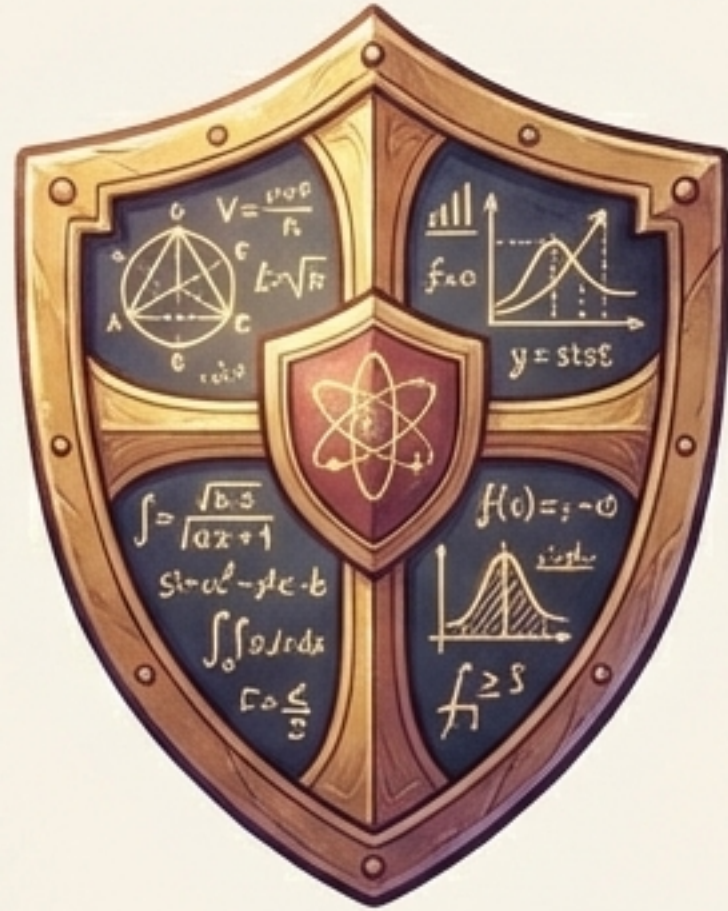
نه فقط کدنویسی در نوت‌بوک.  
باید یاد‌گیری کدهای تمیز و  
ماژولار بنویسی که منطق و فرآیندها رو  
به درستی پیاده‌سازی کنن.



## SQL

توانایی کوئری زدن و استخراج داده از دیتابیس‌ها،  
مهارتی که همیشه و همه‌جا به کارت میاد.

## تجهيزات سفر: زره، سپر و نقشه



### آمار و ریاضیات

این‌ها منطق اصلی و قلب تپنده‌ی مدل‌های یادگیری ماشین و یادگیری عمیق هستند. درک عمیق آن‌ها به تو قدرت شهود می‌دهد.



### Git و Version Control و Git

برای ذخیره کردن روند پیشرفتت، همکاری با بقیه و نگهداری کدها روی پلتفرم‌هایی مثل GitHub. شاید ترسناک به نظر بیاد، اما وقتی یادش بگیری، بهترین دوستت می‌شه.

# هنر کیمیاگری داده: از کاوش تا مهندسی ویژگی

## تحلیل داده اکتشافی (EDA)

اینجا بخش هیجان‌انگیز ماجراست: آزمایش، پاکسازی، تحلیل و مصورسازی داده‌ها. جایی که حس می‌کنی داری پیشرفت می‌کنی و نتیجه‌ی تغییرات رو بلافاصله می‌بینی.



## مهندسی ویژگی (Feature Engineering)

چطور با مقادیر گمشده (missing values) برخورد کنیم؟  
چطور داده‌ها را انکد کنیم (one-hot encoding)؟  
چطور داده‌ها را تقسیم کنیم (split)؟ بر اساس زمان یا تصادفی؟ جواب این سوال‌ها شهود تو را می‌سازد.



نکته حرفه‌ای: کتاب 'Designing Machine Learning' از Chip Huyen از Chip Systems یک گنج واقعی برای ساختن این شهود است.

# جعبه ابزار جادوگری: انتخاب مدل و متریک مناسب مناسب

لازم نیست همه‌ی مدل‌ها رو از بر باشی. با اصول اولیه مثل رگرسیون خطی و لجستیک و مدل‌های درختی (مثل XGBoost و LightGBM) شروع کن. به مرور زمان، شهود پیدا می‌کنی که هر مدل برای چه نوع مسئله‌ای بهتر کار می‌کنه.



متریک‌های ارزیابی، همه چیز هستند! در طبقه‌بندی (classification)، همه سریع به سراغ دقت (accuracy) می‌رن. اما گاهی باید روی Precision یا Recall تمرکز کنی. مثلاً در یک پروژه‌ی تشخیص اسپم، Recall مهم‌تره، چون یک ایمیل اسپم که به اشتباه وارد اینباکس بشه (False Negative)، خیلی بدتر از یک ایمیل سالم هست که اشتباهی به پوشه‌ی اسپم بره.

## فراتر از کگل: به دنیای واقعی خوش آمدید

کگل برای تمرین عالی. داده‌ها تمیز و آماده هستن و معمولاً حجم زیادی ندارن. این یک محیط کنترل‌شده و خوب برای شروع.

اما در دنیای واقعی، تو باید خودت مسئله رو پیدا کنی، داده‌ها رو از منابع مختلف جمع‌آوری کنی، تمیزشون کنی و ساختار بدی. باید خلاق باشی. سعی کن پروژه‌هایی رو انجام بدی که مدل رو به اینترنت متصل کنی تا هر کسی بتونه ازش استفاده کنه. اینطوری از یک دموی باحال، به یک محصول قابل استفاده می‌رسی.

# دره‌ی مهندسی: جایی که نفس خیلی‌ها می‌گیرد

وقتی اسم کانتینر، CI/CD و تست‌نویسی میاد، خیلی از دانشمندان داده جا می‌زنن. منم دقیقاً همینطور بودم. این مفاهیم هیچ شباهتی به دنیای رنگارنگ مصورسازی داده و بالا بردن دقت مدل ندارن.

اما حقیقت بی‌رحمانه اینه: بدون این قدم‌ها، تمام زحمات تو در نوبت بوک تو دفن می‌شه و ارزشی که ساختی، دقیقاً صفره. چون هیچ کاربری هیچوقت ارزش استفاده نخواهد کرد.

# ساختن پل روی دره‌ی مهندسی

## Docker (کانتینرها):

پروژه‌ها را بسته‌بندی می‌کند تا همه جا یکسان اجرا بشه. دیگه خبری از «روی سیستم من کار می‌کرد!» نیست. محیط اجرای تو، با همکاری یکی می‌شه.

## Unit Tests (تست نویسی):

در هر مرحله‌ی مهم از کد، چک‌پوینت‌هایی قرار می‌دی تا مطمئن بشی همه چیز درست کار می‌کنه و با داده‌های عجیب و غریب یا outlierها غافلگیر نمی‌شی.

## (CI/CD GitHub Actions)

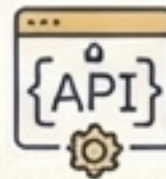
فرآیند تست و استقرار کدها را اتوماتیک می‌کند. با هر تغییر، تست‌ها خودکار اجرا می‌شن و همه چیز برای رفتن روی کلاود آماده می‌شه.

# لحظه‌ی پرتاب: وقتی مدل شما به دنیا سلام می‌کند



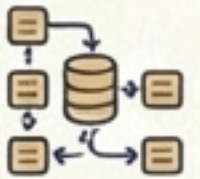
## ساخت API

با ابزارهایی مثل FastAPI یا Flask، برای مدل خودت یک درگاه ورودی (API) می‌سازی. اینطوری مدل تو از طریق وب (HTTP) قابل دسترس می‌شه و می‌تونه درخواست‌ها رو جواب بده.



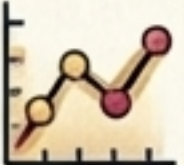



## انتخاب استراتژی

- پیش‌بینی دسته‌ای (Batch Prediction): داده‌ها در بازه‌های زمانی مشخص (مثلاً هفته‌ای یک‌بار) می‌رسن و مدل روی کل دسته اجرا می‌شه.
- پیش‌بینی آنی (Real-time Prediction): داده‌ها به صورت جریانی و لحظه‌ای وارد می‌شن و مدل باید فوراً پاسخ بده.



# نگهبانی از قله: مانیتورینگ و کنترل ورژن

کارت با دیپلوی تموم نمی‌شه! مدل‌ها در طول زمان دچار افت عملکرد (drift) می‌شن چون داده‌های دنیای واقعی تغییر می‌کنه. باید همیشه حواست به این چیزا باشه:

- مانیتورینگ: رصد کردن معیارهایی مثل data drift و latency. 
- ورژن کنترل کد (با Git): همیشه می‌دونی چه کدی در حال اجراست. 
- ورژن کنترل مدل (با MLflow): «شاید یک روزی به مدلی که یک ماه پیش با هایپرپارامترهای خاصی ساختی نیاز پیدا کنی. اگه نتونی پیدا کنی، برای همیشه از دست رفته.» 
- ورژن کنترل داده: بدونی دقیقاً چه داده‌ای برای آموزش هر ورژن مدل استفاده شده. 



امروزه بیشتر شرکت‌ها از سرویس‌های ابری استفاده می‌کنند. لازم نیست یک معمار راهکارهای ابری (Solutions Architect) بشی، اما باید با اصول اولیه یکی از این پلتفرم‌ها آشنا باشی.

# آشنایی با سرزمین ابرها: AWS, GCP, Azure



- چطور داده‌ها رو در سرویس‌هایی مثل S3 ذخیره کنی.


- چطور با سرویس‌های یادگیری ماشین مثل AWS یا SageMaker کار کنی.


- چطور کانتینر داکر خودت رو روی سرویس‌هایی مثل ECS اجرا کنی.

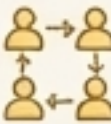
یکی رو انتخاب کن و چند تا پروژه واقعی روش انجام بده تا دستت راه بیفته.

# اکسیر موفقیت: مهارت‌های نرمی که تو را متمایز می‌کند

تو می‌تونی ماهرترین متخصص فنی دنیا باشی، اما اگه در یک شرکت فعال در حوزه معدن طلا کار کنی و از صنعت و KPIهای اوها هیچی ندونی، فلج می‌شی.

• **درک عمیق از کسب‌وکار و صنعت: بفهمی که شرکت چه چیزی برایش مهمه.** 

• **ارتباط موثر: تو همیشه پشت کامپیوتر قایم نمی‌شی. باید با مدیران، همکاران و کاربران نهایی صحبت صحبت کنی تا نیازهاشون رو بفهمی.** 

• **همکاری تیمی: علم داده یک ورزش تیمی است.** 

هدف، فقط بالا بردن دقت مدل نیست. هدف، حل کردن یک مشکل واقعی برای یک انسان واقعی است.



# نقشه گنج تو آماده است: این هم اولین قدم



جنگل تجهیزات

Python

SQL

Git

دنیای نوت بوکها

کارگاه کیمیاگری

کارگاه کیمیاگری

دره‌ی مهندسی

دره‌ی مهندسی

قله‌ی استقرار

این نقشه، مسیر کامل سفر توست. از آماده کردن تجهیزات تا رسیدن به قله‌ی پروداکشن. برای اینکه این مسیر انتزاعی نباشه، می‌تونم یک پروژه‌ی کامل و جامع رو از اول تا آخر دنبال کنی.



من یک پروژه‌ی یک ساعت و نیمه برای پیش‌بینی ریزش مشتری (Churn) انجام دادم که تمام این مراحل رو پوشش می‌ده:

- تحلیل داده و ساخت پایپ‌لاین‌ها
- استفاده از MLflow برای ردیابی آزمایش‌ها
- بسته‌بندی با Docker
- اتوماسیون با GitHub Actions (CI/CD)
- استقرار نهایی روی AWS

شروع اولین ماجراجویی