Building Efficient Recommender Systems: Techniques, Challenges, and Best Practices

KATERINA ILIAKOPOULOU-ZANOS

Machine Learning Software Engineer
Meta

Recommender Systems are everywhere

FACEBOOK

INSTAGRAM

GOOGLE

AMAZON

SPOTIFY

THE NEW YORK TIMES

PINTEREST

TIKTOK

TINDER

NETFLIX

MICROSOFT

BUZZFEED

15%

of businesses' ML projects scale across multiple parts of the business.

Source: *Operationalizing machine learning in processes*

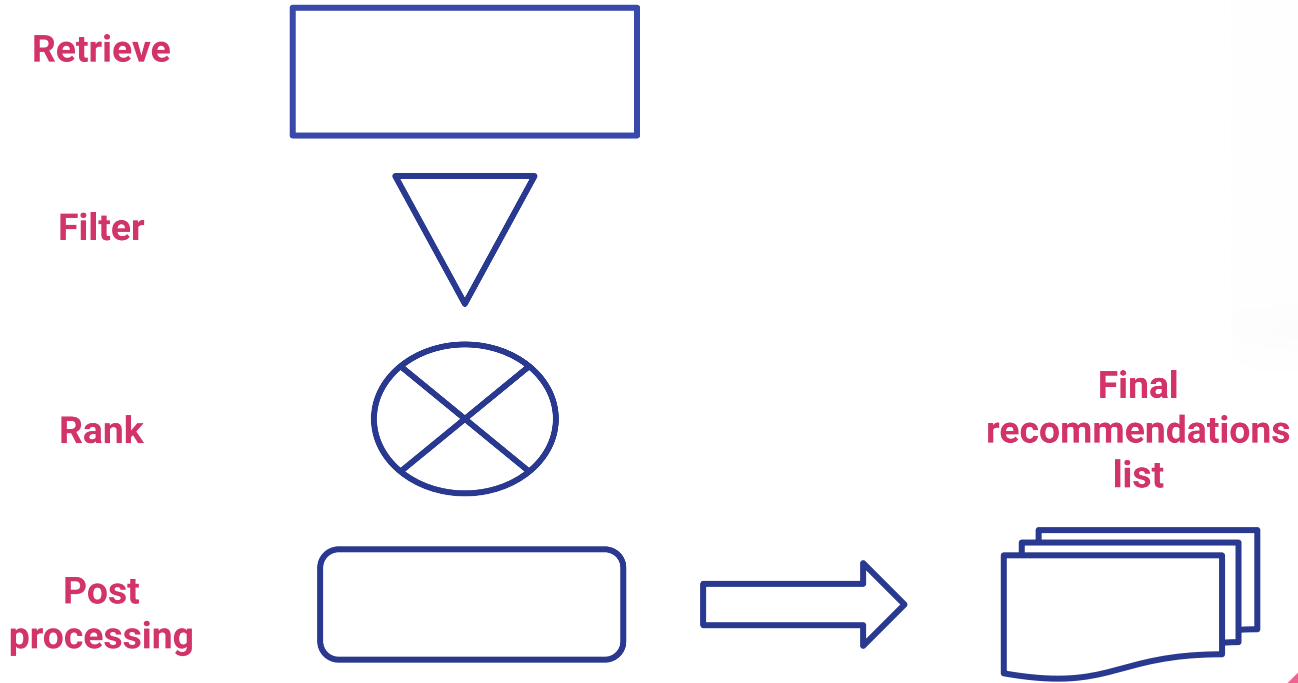
87%

of machine learning models never make it into production.

Source: *Why do 87% of data science projects never make it into production?*

Building Recommender Systems Efficiently

What is a recommender system



Choosing the right model

Trending / Most Popular

Collaborative Filtering (User-User, Item-Item, User-Item)

Content Based

Contextual Bandits

Deep Learning

Get the data

User History / Engagement data

Precomputed features

Embeddings

Evaluation

Offline Evaluation

- Recall
- Precision
- F1 Score
- Calibration

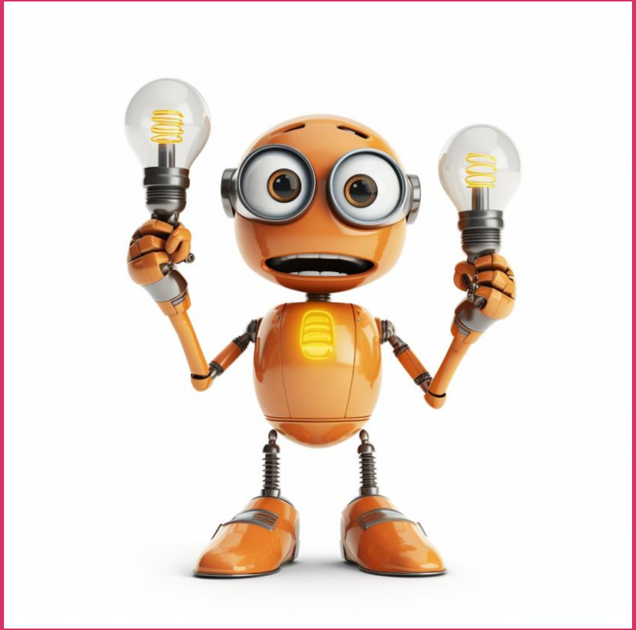
A/B Testing

- Stat-sig results on core business value metrics

Building efficient recommender systems

1. **Responsiveness**
2. **Speed of development**
3. **Cost of deployment**

Responsiveness



[**Responsiveness**] How fast can we fetch fresh content to recommend to the user?

Indexing content and extracting features as soon as it becomes available

Identify potential bias in the model that could punish new content

Leverage exploration / exploitation techniques to allow for gathering the right data for recommending content to the right audience

Choosing between offline & online inference

[**Responsiveness**] How fast can we deliver recommendations to the user?

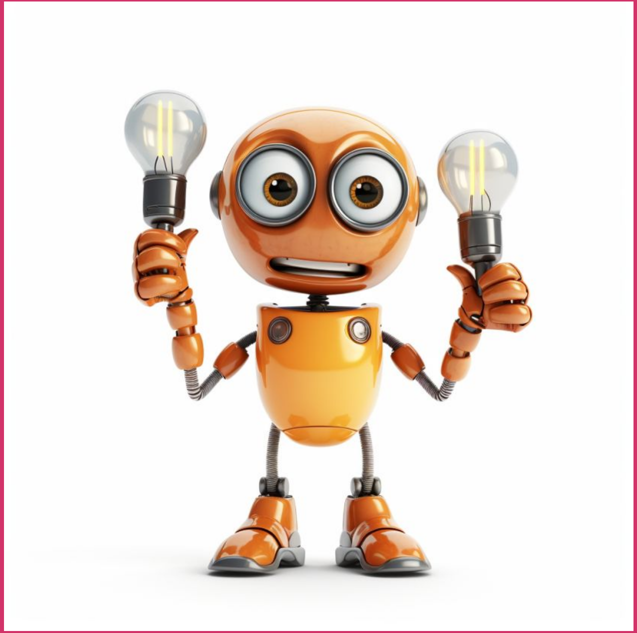
Cache

Replication for handling multiple requests

Optimizing data requests

How our system needs to respond defines key architecture & modeling decisions

Speed of development



[Speed of Development] How do you align work and priorities with your team and partner teams?

How big is your team? Are you working with ML Engineers, Data Scientists, Data Engineers?

Getting features / data ready for training the model and later for inference

Working with partner teams on conflicting goals

[Speed of Development] Working with devops on architecting the desired solution

How many systems do we have to touch in order to make our solution production ready?

Are the changes simple and straightforward or do we have to build new functionalities to make our solution available?

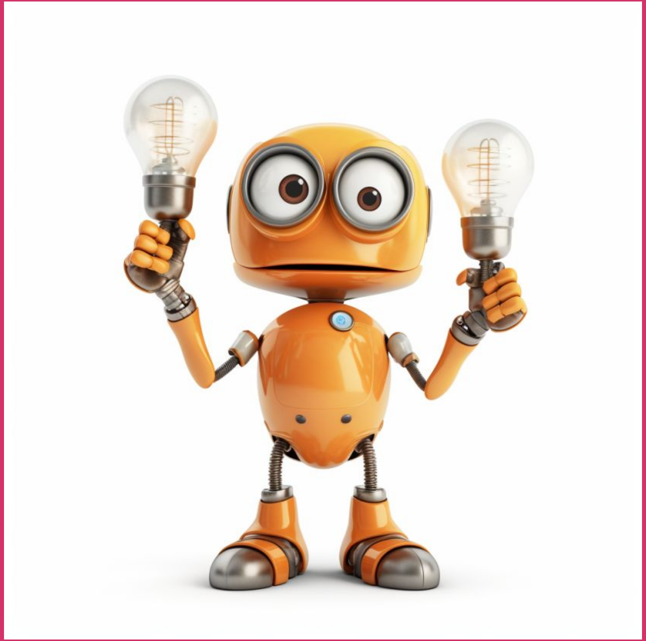
[Speed of Development] How can we test early and have results to know if a recommender solution is effective for our product?

Do upfront analysis to prove that the recommender solution could work well with your data before moving to online testing

Test gradually. Develop an MVP with minimum features before jumping into a more complex system

Identify internal and external dependencies to prioritize efficiently.
Test often - both offline and online.

Cost of deployment



[Cost of Deployment] What are we willing to sacrifice over achieving desired success goals?

CPU vs Storage Space

DB responsiveness and availability

Data & Model cost

Work hours

[Cost] What is the accepted cost of my recommender system?

What do we gain with the product we are shipping?

If we prompt for a cheaper solution how much does it hurt our user value?

If we achieve the desired trade off now, how fast is this bound to change as our user base expands?

If possible, decide on what your cost ceiling is first. It will set the parameters of your system later.

Building efficient recommender systems

1. **Decide how the recommender system drives user value.**
2. **Set an upper limit for system cost.**
3. **Make architecture decisions for the system and model based on how responsive we want it to be in terms of content delivery and freshness.**
4. **Optimize for speed of development by identifying key dependencies and aligning on priorities with partner teams.**
5. **Assess the cost on resources and choose the best tradeoff.**
6. **Agree on launch criteria within your org.**
7. **Test gradually and often as you develop the system.**
8. **Plan for scaling the solution to a larger user base.**

Thank you!

LinkedIn:

https://www.linkedin.com/in/katerinailiakopoulou/

Twitter: @matzika00

https://www.katerinazanos.com/