TADA : Talk About Data Analytics & Science l Week 17

Ranking Algorithms

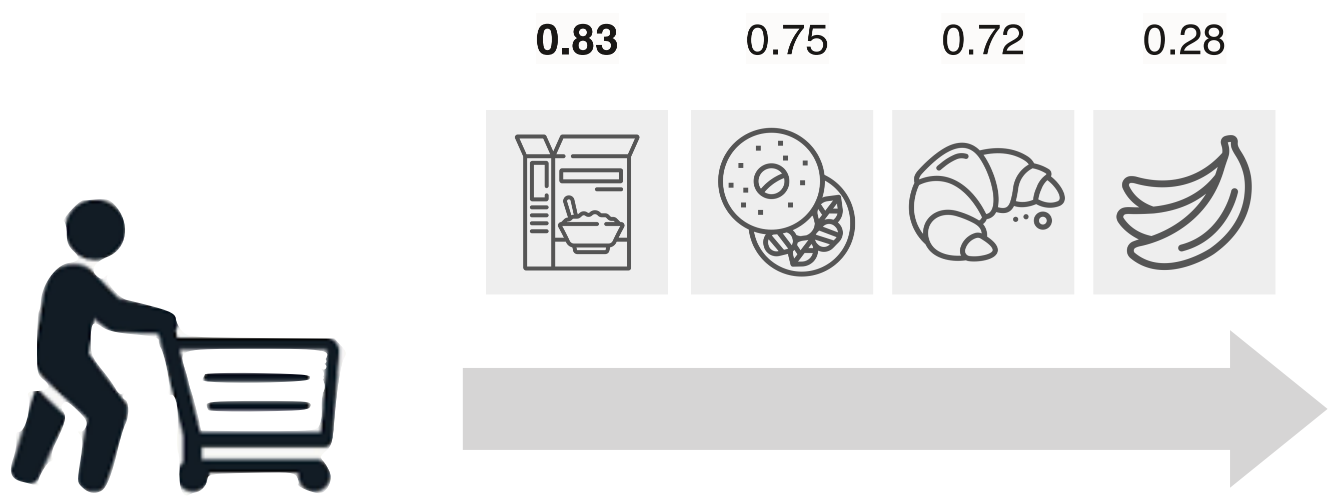
in Machine Learning & Deep Learning

Hannah Do l April 23rd, 2022

Contents

1. Introduction

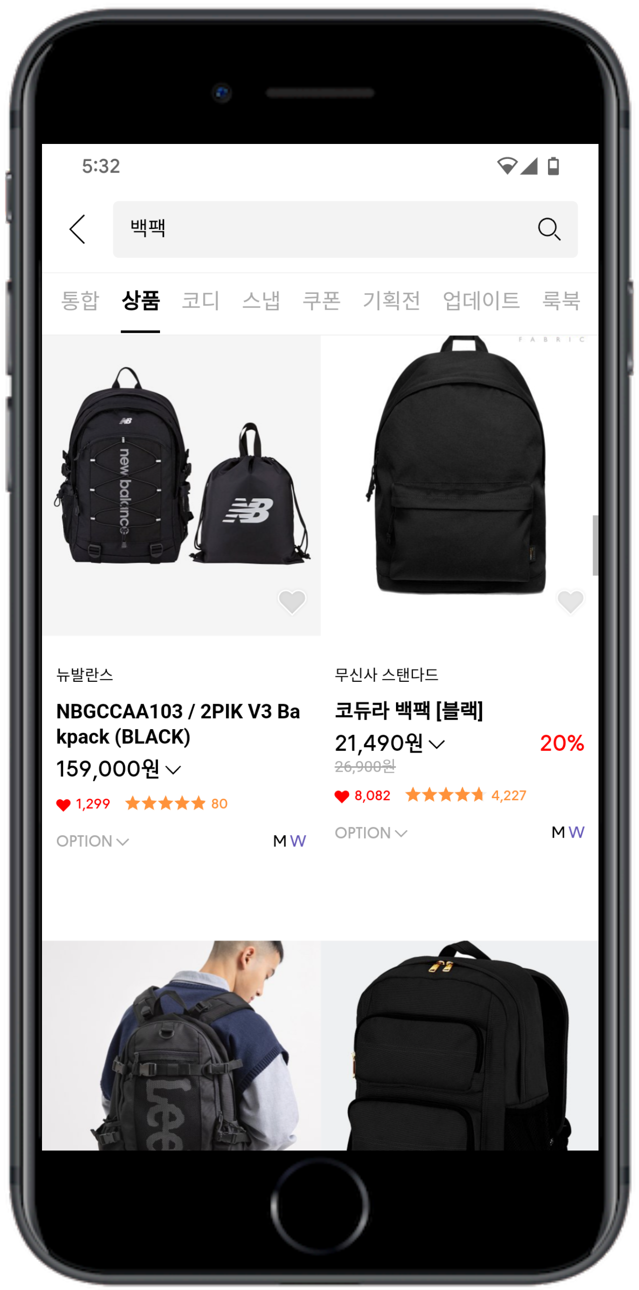
Ranking is a ML/DL technique **to rank items**, commonly used in fields that require information retrieval and display - such as E-commerce, Social Networks, Recommendation Systems, and so on.



"For an online shopper, it becomes important that similar items of relevance appear to the user such that the user clicks or purchases the item."

The ranking technique directly ranks items by training a model to predict the ranking of one item over another item by creating a "score" for each item.

A. Learning To Rank (LTR) in Search Engines



Three major approaches to LTR

i) Pointwise :

Compares single item at a time to discover best ranking for independent item. (Computing *relevancy* for the item)

- Classification → Gathers similar items in the same class or category.
- Regression → Gives similar items a similar function value - to assign similar preferences for the ranking procedure. (Ex. Logistic Regression)

A. Learning To Rank (LTR) in Search Engines

Three major approaches to LTR

ii) Pairwise :

Compares two items together as a pair, higher-lower pairs are adjusted in ranks. The goal is to minimize number of cases where the pair of results are in the wrong order. Predicting relative order is closer to the nature of ranking than predicting class label or relevance score.

"Training a classifier to predict if $r_u < r_v$ based on pairs of training documents with the same query"

Ranking Algorithms developed by researchers at Microsoft Research

1) RankNet :

Uses Mini-batch Stochastic Gradient Descent to update learning function and requires a differentiable function such as that used in neural networks.

$$P(y_{u,v} \mid x_u, x_v) = \frac{1}{1 + e^{f(x_u) - f(x_v)}}$$

Where $y_{u,v}$ as a binary value (0, 1), $f(x) = \theta$ as a logistic regression function

2) LambdaRank

Gradient estimated from the candidates pair is defined as lambda, and scaled by a metric called nDCG.

3) LambdaMART (Also part of Listwise approach)

Based on MART (Multiple Additive Regression Trees) that exploits gradient boosted trees where the gradient is computed after each new tree and to estimate the direction that minimizes the loss function. LambdaMART uses this ensemble of trees however replaces the gradient with the lambda (gradient computed given the candidate pairs).

A. Learning To Rank (LTR) in Search Engines

Three major approaches to LTR

iii) Listwise :

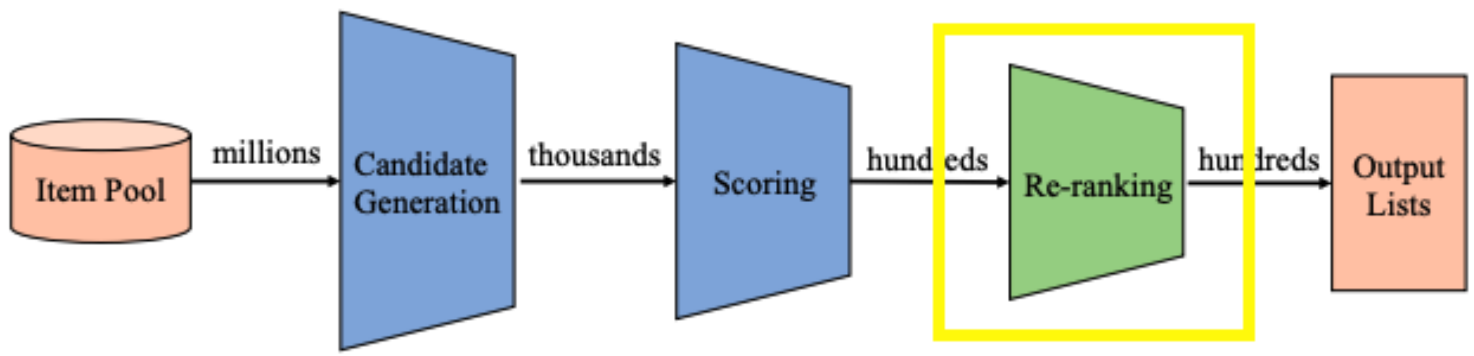
Decides the optimal ordering of an entire list of items. Probability model is often used to minimize ordering error. NDCG

1. Direct optimization of IR measures such as NDCG (ex. SoftRank, AdaRank)
2. Minimizing a loss function based on unique properties of the kind of ranking (ListNet, ListMLE)

Currently available open source solutions

- Elasticsearch (sltr)
- Solr
- Algolia
- Elastic Cloud
- Amazon Cloud Search
- Azure Search
- RankLib (The Lemur Project)

B. Re-ranking in Recommendation Systems



Why Re-rank after scoring?

1. Reduce Cost - a simple similarity metric is used to retrieve an initial set of relevant documents (top-k retrieval) and then ML/DL algorithms are used to re-rank the initial set of k documents.
    - Simple ranking function such as BM25 (Okapi Best Matching 25 : Bag-of-Words retrieval function) is used to rank a set of documents based on the query terms appearing in each document.
2. Add Recent Data
    - What kind of items has the user been recently looking for?
    - What kind of links clicked on?
    - How long did the user stay on that link?
    - Friends/families interests and clicks
    - Location
    - Native Language
    - Age
3. Add Diversity
    - Recommending similar items only may result in customers losing interest
    - Ex. YouTube algorithm recommends new or random video on purpose once in awhile

B. Re-ranking in Recommendation Systems

Research Papers on Re-ranking Algorithm for Recommendation Systems

1. Statistical Approach

- Personalizing Fairness-aware Re-ranking (Liu & Burke, Cited by 33, 2018)
  - P-fairness threshold

- Managing popularity bias in recommender systems with personalized re-ranking (Abdollahpouri et al., Cited by 112, 2019)
  - Method to include “Long Tail” via xQuAD-based Re-ranking Algorithm

- Item Re-ranking methods for Recommender Systems (Adomavicius & Kwon, Cited by 65, 2009)
  - Trade Off between Accuracy and Diversity
  - Pareto Principle : Controlling the threshold between the top 20% popular items with the other 80%.

2. ML/DL Approach

- Personalized Re-ranking for Recommendation (Pei et al., Cited by 59, 2019)
  - Input, Encoder (Self-attention), Output - known as **PRM**

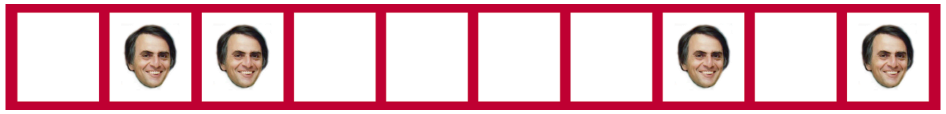
- Personalized Re-ranking with Item Relationships for E-commerce (Liu et al., Cited by 10, 2020)
  - GNN model that replaces the flaws in RNN, SA
  - Item Relationship Graph Neural Networks for Personalized Re-ranking (IRGPR)
    - Item relationship graph and User-item scoring graph combined to maintain complementarity, substitutability

3. Evaluation Methods in Information Retrieval System

i) Precision@k

- Percentage of the items that are relevant.

$$\text{Precision@10} = \frac{\text{Relevant Items}}{\text{Viewed Items}} = \frac{n}{k} = \frac{4}{10}$$



ii) nDCG :

- Normalized Discounted Cumulative Gain
- DCG : The gain is accumulated from the top of the result list to the bottom, with the gain of each result discounted at lower ranks.
- $\mathrm{nDCG_p} = \frac{DCG_p}{IDCG_p}$ where IDCG is ideal discounted cumulative gain : $\mathrm{IDCG_p} = \sum_{i=1}^{|REL_p|} \frac{rel_i}{\log_2(i+1)}$

iii) MRR :

Mean Reciprocal Rank,
The reciprocal rank of a query response is the multiplicative inverse of the rank of the first correct answer: 1 for first place, ½ for second place, ⅓ for third place and so on. The mean reciprocal rank is the average of the reciprocal ranks of results for a sample of queries Q:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}.$$

Query	Proposed Results	Correct response	Rank	Reciprocal rank
cat	catten, cati, **cats**	cats	3	1/3
torus	torii, **tori**, toruses	tori	2	1/2
virus	**viruses**, virii, viri	viruses	1	1

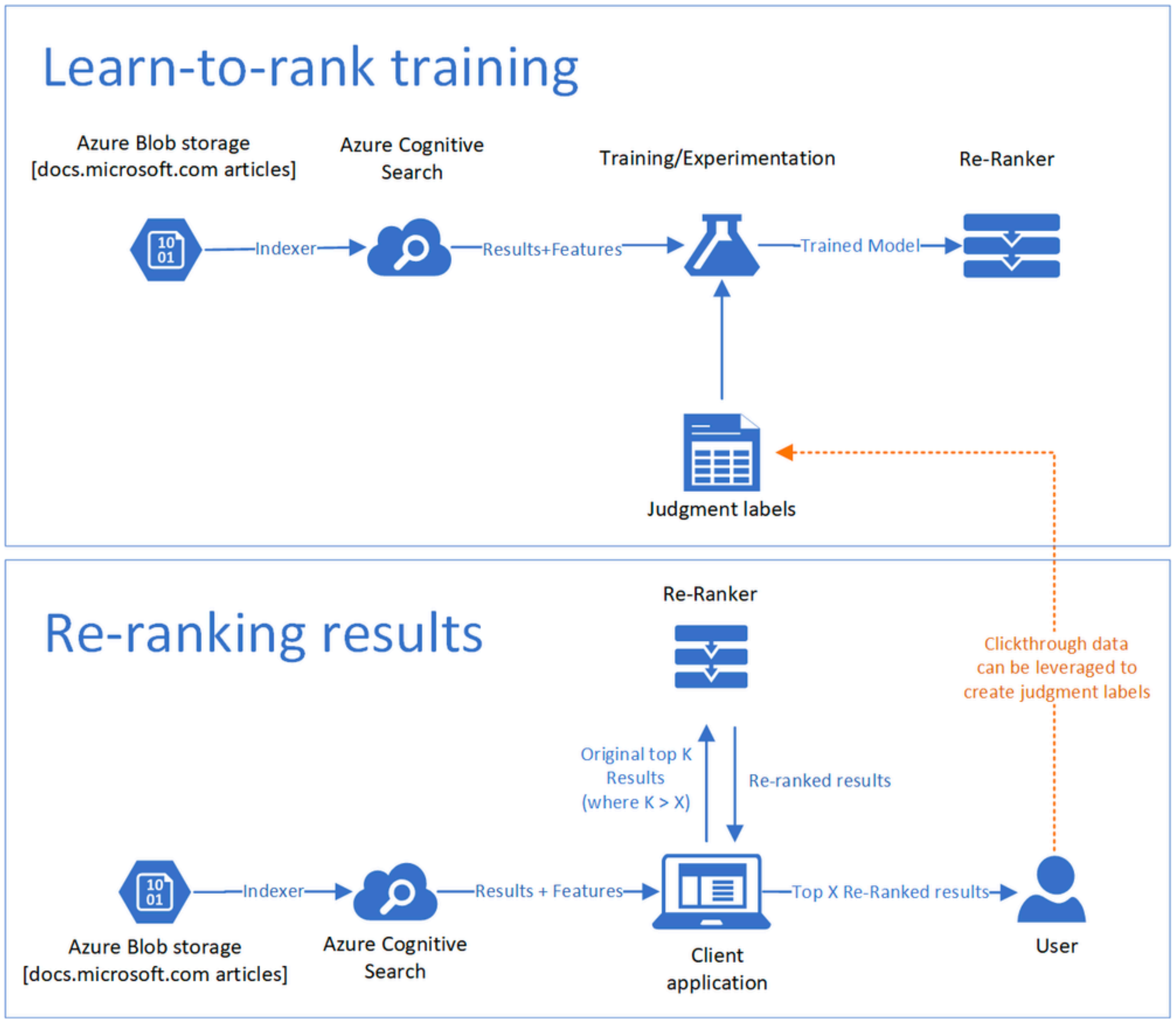
Given the three samples, we could calculate the mean reciprocal rank as (1/3 + 1/2 + 1)/3 = 11/18 or about 0.61.

iv) MAP :

Mean Average Precision : AP averaged across multiple queries
AP : Average precision over ranks 1…k for each k where rank k item is relevant

4. Case Study

A. Microsoft



Microsoft : Azure Cognitive Search LTR Process

Re-ranking via **XGBRanker** with k-fold training steps

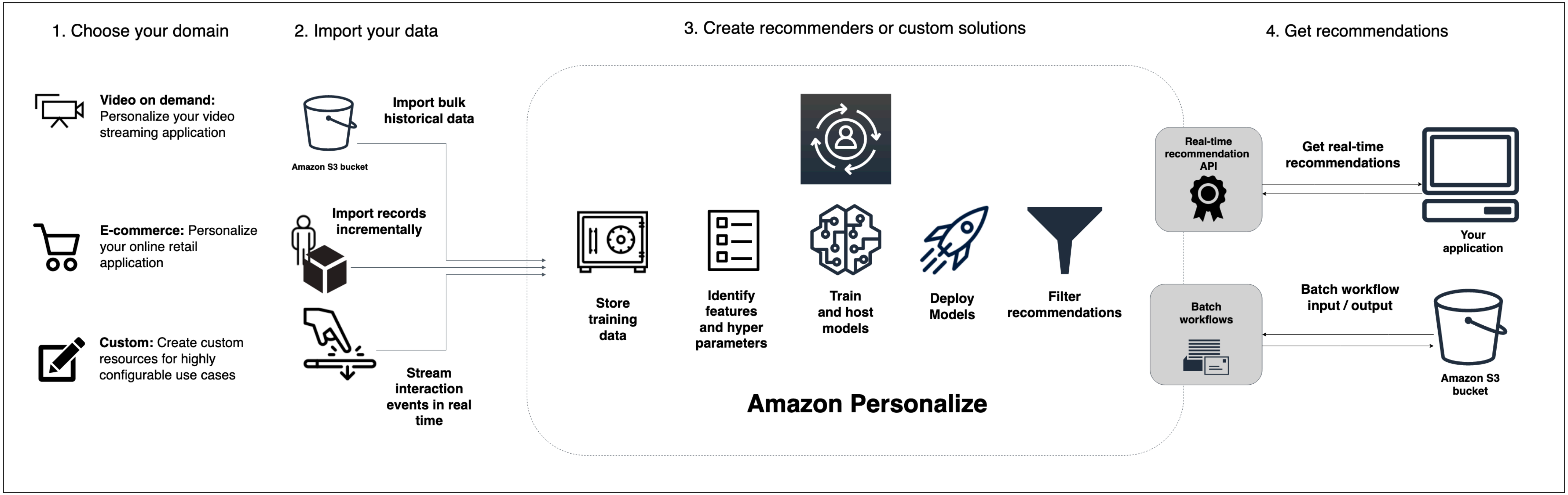
```
import xgboost as xgb

params = {'objective': 'rank:ndcg', 'learning_rate': 0.1,
          'gamma': 1.0, 'min_child_weight': 0.1,
          'max_depth': 8, 'n_estimators': 500}

ranker = xgb.XGBRanker(**params)
ranker
```

B. Amazon Personalize

Personalized-Ranking :

Amazon Personalize can blend real-time user activity data with existing user profile and item information (historical data) to recommend the most relevant items for the user.

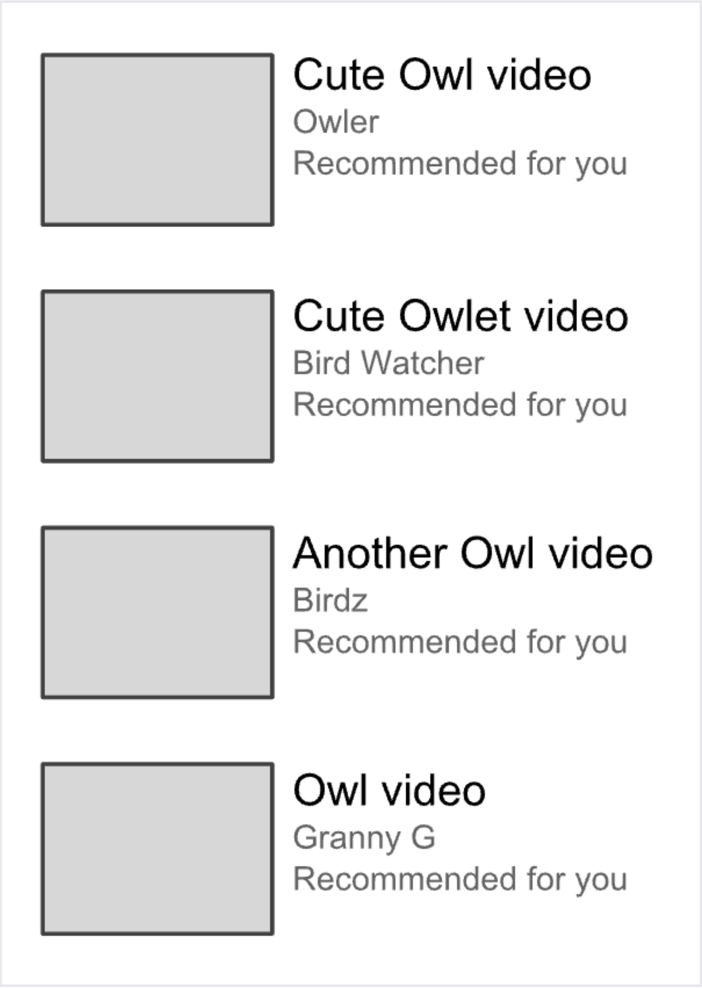
Features used : User’s current location / device type (contextual metadata), list of items to rank (inputList), campaignArn (ARN of campaign to be used, ex. **HRNN**), …



Amazon > Amazon Personalize Runtime > GetPersonalizedRanking

Re-ranks a list of recommended items for the given user.

C. Google Developers (Youtube)



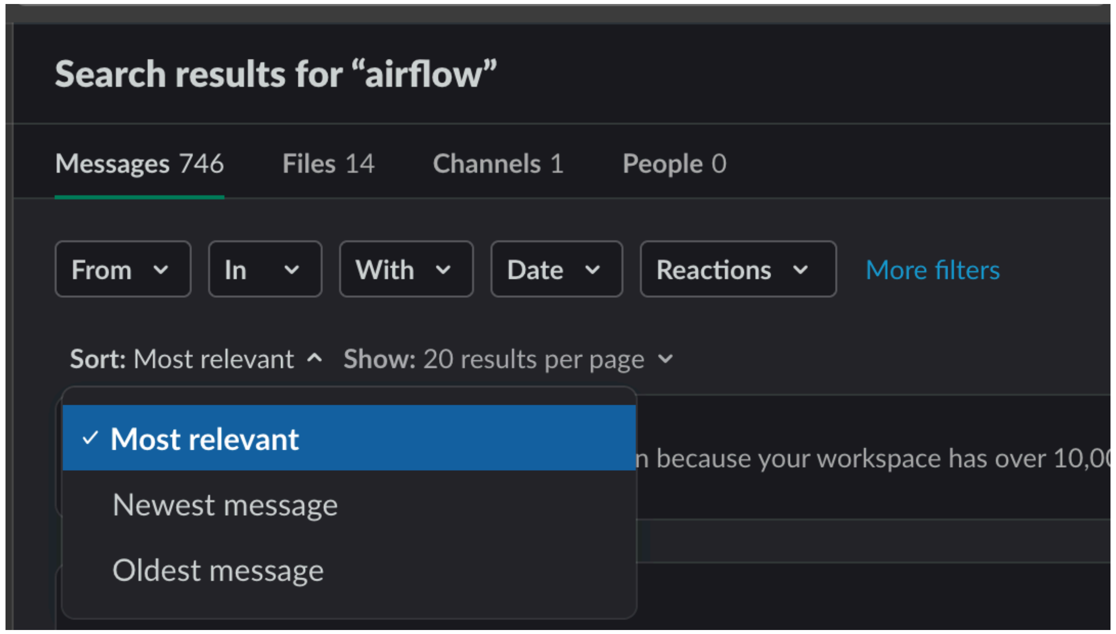
Re-ranking Algorithm used in Youtube

1. Using filters that remove certain candidates (clickbait)
2. Modifying the score returned by the ranker using features such as the age or length of the video.
    1. Freshness : Uses the most recent data, warm start > cold start (saves training time), **DNN** (softmax) model, selected features such as video's age, recently viewed hour, etc.
    2. Diversity : User can lost interest if recommended videos have similar pattern. Re-ranking's purpose can be to include different item once in awhile.
        - Preparing diverse dataset
        - Training separate re-ranking models with different purposes
    3. Fairness : Keeping the training dataset as fair as possible. Continuous evaluation of biases in different demographic data is required, training a model separately for minority group may be necessary

D. Slack

Recent and **Relevant** Search

Uses Solr's custom sorting functionality and focuses on a labeling strategy that judges the relative relevance of documents within a single search using clicks known as a Pairwise Transform.



SparkML's built-in **SVM** algorithm based on Query, Present (recent) ranking, and the additional features.

Additional Features : age of the message, Lucene score of the message, priority score of the searcher's DM, whether the message was pinned, had emoji reactions, etc.

Slack exposed the machine-learned re-ranker for **Relevant** sort to 50% of users. "For our top-line metrics, we looked at sessions per user, searches per session, clicks per search, and click-through rate among the top 1, 2, and 5 search results. As previously mentioned, we saw significant gains over the existing *Relevant* search — a 9% increase in clicked searches and among searches that received at least one click, a 27% increase in clicks at position 1."

Concluding remarks

LTR's ranking algorithm is thoroughly researched, however when it comes to re-ranking in recommendation algorithms, the implementation methods and applications differ depending on the **purpose** of the re-ranking algorithm.

1. What is the main purpose of your re-ranking algorithm?
   - Cost Reduction
   - Diversity
   - Accuracy / Adding recent information
2. Statistical Re-ranking vs. ML/DL Re-ranking
   - ML/DL models may not be necessary depending on the purpose of the re-ranking algorithm.

5. References

- “Ranking” Oracle Machine Learning Database https://docs.oracle.com/en/database/oracle/machine-learning/oml4sql/21/dmcon/ranking.html#GUID-ACD212DD-F13F-40E5-B365-15DB63C55B05

- “The ABCs of Learning to Rank” James Le, Lucidworks (Nov 1st, 2019) https://lucidworks.com/post/abcs-learning-to-rank/

- “Re-ranking Cognitive Search results with Machine Learning for better search relevance” Luis Cabrera-Cordon, Microsoft (Jul 23rd, 2020) https://techcommunity.microsoft.com/t5/ai-cognitive-services-blog/re-ranking-cognitive-search-results-with-machine-learning-for/ba-p/1542431

- “GetPersonalizedRanking” AWS Documentation, https://docs.aws.amazon.com/personalize/latest/dg/API_RS_GetPersonalizedRanking.html

- “Re-ranking” Google Developers, https://developers.google.com/machine-learning/recommendation/dnn/re-ranking

- “LETOR: A benchmark collection for research on learning to rank for information retrieval” Tao Qin, Tie-Yan Liu, Jun Xu, Hang Li (April 29th, 2009) https://www.microsoft.com/en-us/research/wp-content/uploads/2016/08/letor3.pdf

- “Search at Slack” Isabella Tromp, John Gallagher, Jason Liszka (2016) https://slack.engineering/search-at-slack/