Flipkart

GRiD 5.0

Problem Statement Title: **Personalized Product Recommendations**

Team Name: ***686157-U8TFL296***

Team members details

Team Name	686157-U8TFL296		
Institute Name/Names	Indian Institute Technology Bhilai		
Team Members >	**1 (Leader)**	**2**	**3**
Name	Dakshraj Sadashiv Kashid	NA	NA
Batch	2020 (Admitting) 2024 (Graduating)	NA	NA

Deliverables/Expectations for Level 2 (Idea + Code Submission)

Problem Statement: **Personalized Product Recommendations**

- Code is submitted along with the implementation detail.
- The description of idea is mentioned in the following slides.

Glossary

- **DLRM:** Deep Learning Recommendation Model by Facebook

- **NDCG:** Normalized Discounted Cumulative Gain, it accounts for ranking of relevant items to its respective position.

- **MLP:** Multi-Layer Perceptron.

- **FM:** Factorization Machine.

Instructions (You Can Delete this Slide)

Dear Team,

Congratulations on reaching this stage - We look forward to some amazing & innovative solutions.

Please find some important instructions before you begin to prepare your submission decks.

Slide Limit : 10 Slides of Content **post (after)** this Slide
Saving Format : Save the file as a PDF to ensure your formatting remains intact
Submission Guide: Only the **'Team Leader'** will be able to submit the Deck.
Only the latest submission will be considered as final
(You can keep updating your deck within the deadline)

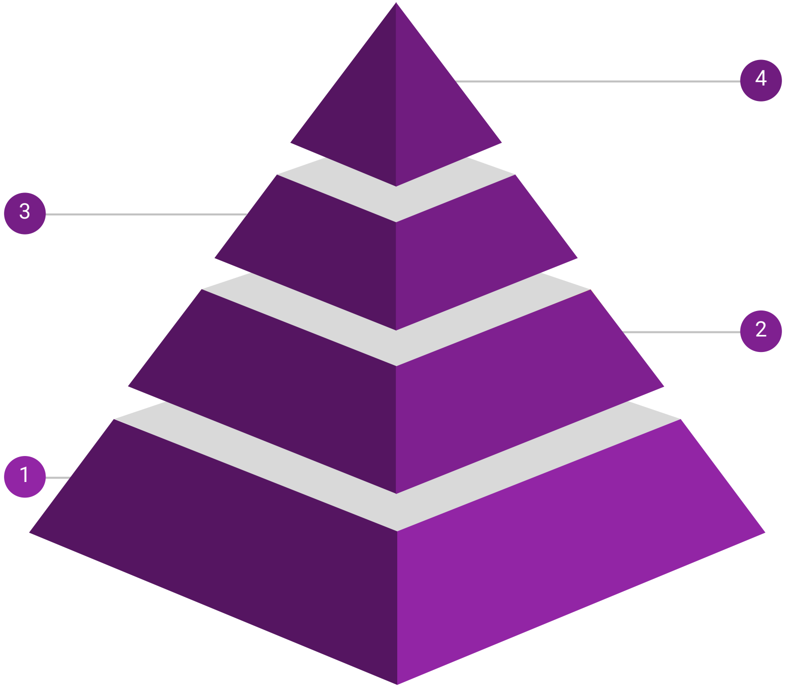
Wishing you all the very best !

Team Flipkart GRiD

Use-cases

- Personalized product recommendations based on
  - user-product interaction
  - product-product relation
  - user-user similarity.
- For ecommerce business having large number of unique products.
- With massive user-interaction providing low latency solution for dynamic product recommendation.

Solution statement: Brief Overview of Multi Stage Product Recommendation



Candidate Retrieval:
Getting top k candidates for the recommendation based on user-item interaction.

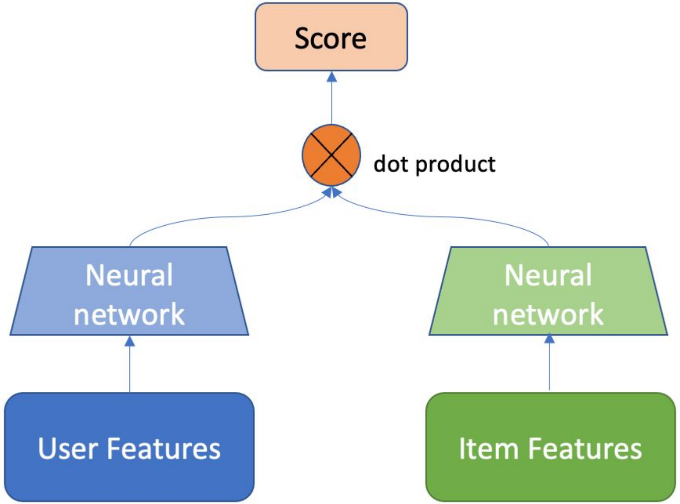
Filtering
Filtering out the retrieved candidates which are redundant.

Ranking
Using deep learning on the reduced data to score the candidates.

Ordering or Categorizing
Ordering based on category, applying business filter.

Candidate Retrieval

- Computational load of deep learning model increases with the number of unique products.
- Used **Two tower model** to get the top k-candidates.
- It captures the product and user interaction by computing the dot product of shared embedding space.



Filtering

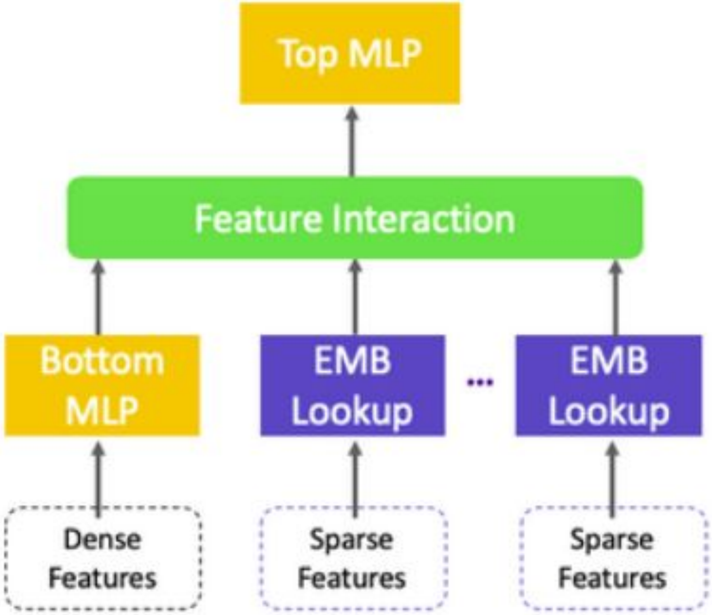
- In user-product interaction recent interaction will yield be better results as compared to previous ones. *(filtered using timestamp of the session)*

- Each user will interact with relatively low number of products as compared to entire product space resulting in sparse interaction matrix.

- I consider only positive interactions and apply negative sampling to counter this problem.

Ranking Stage

I used DLRM model developed by Facebook for the ranking of the products obtained from the candidate retrieval stage.



- Uses embeddings to process sparse features and MLP(*bottom)* for dense features.
- Then a FM calculate feature interaction for sparse setting.
- Then a MLP *(top)* computes the ranking for the products by capturing all the interaction.

Ordering (Business Filtering)

- From the ranked product filter the product based into various business category.

- Various filters:
    - Excluding the out of stock products from the result.
    - Including exclusive offers.

Note: This is currently not implemented into the solution.

Data Preparation

I used the synthetic data generator and created following data:

- User demographic data
- Product description data
- User-Product CTR interaction.

Performance:

For measuring performance I used *Recall@10* and *NDCG@10*.

- Recall@10: How many relevant items were found from the expected items.
- NDCG@10: Score of recommended item as compared to expected ordering of recommendations.

Limitations

- The ranking stage would have tradeoff between latency and quality of recommendation.
    - As top MLP would require wide features which can become a computational bottleneck.

- To capture the similarities between user, I used demographic data.Either it has to be captured during user login or would not be available.

- For the deployment we have to store the feature embeddings for both user and items to provide dynamic recommendation. This has to be updated on regular basis.

Future Scope

- The models and computation can be hardware accelerated using GPU architecture.
- Incorporating time series component to handle the seasonality component of the purchase.
- Separating the ranking stage for each business filter component such as trending, new-arrival, past-purchase and many more. As each of them can have different interaction basis.

Flipkart

GRiD

5.0

Thank You