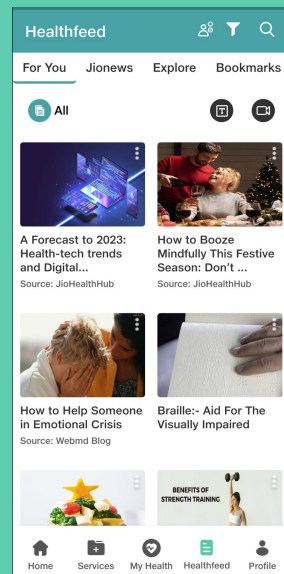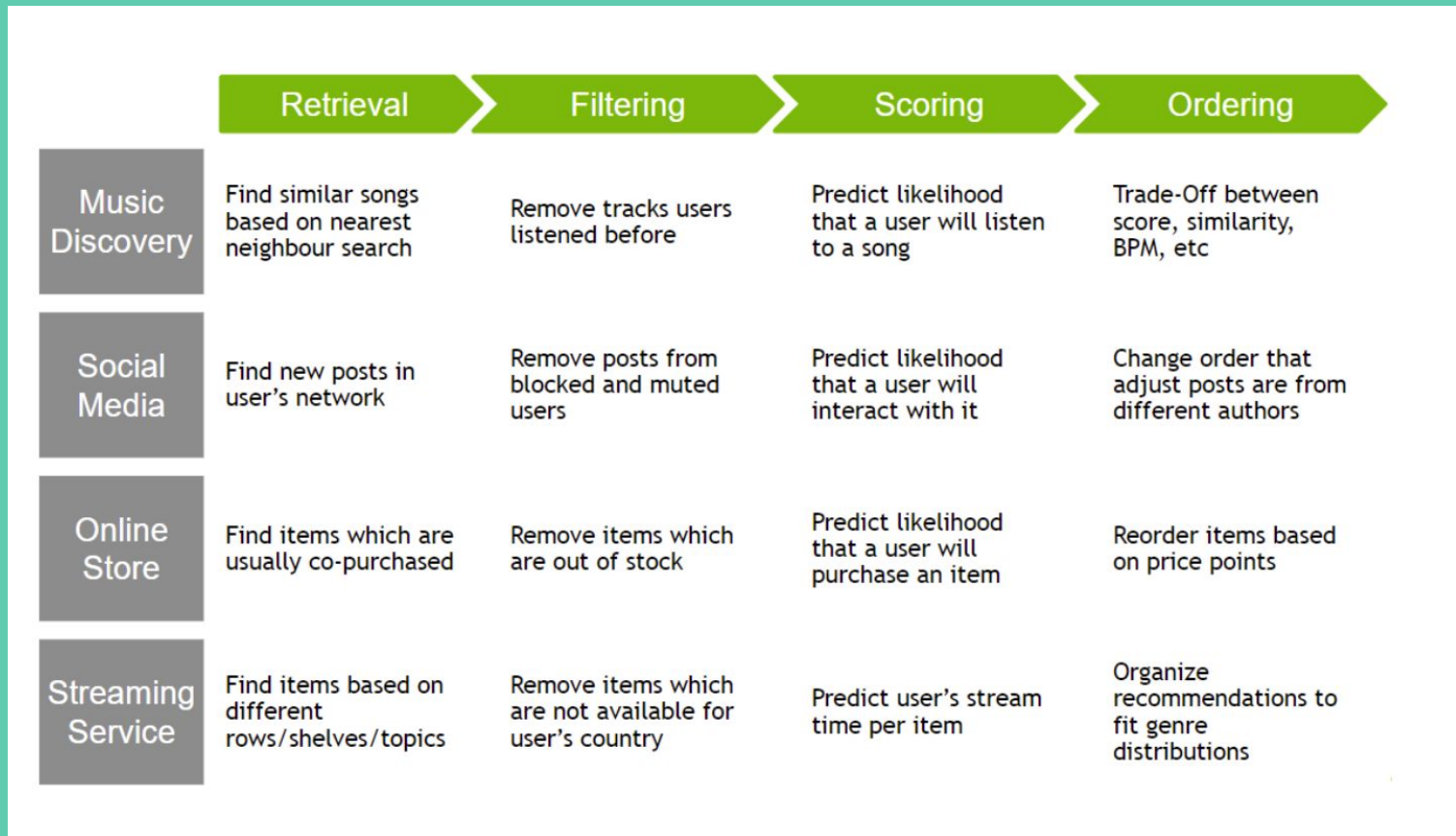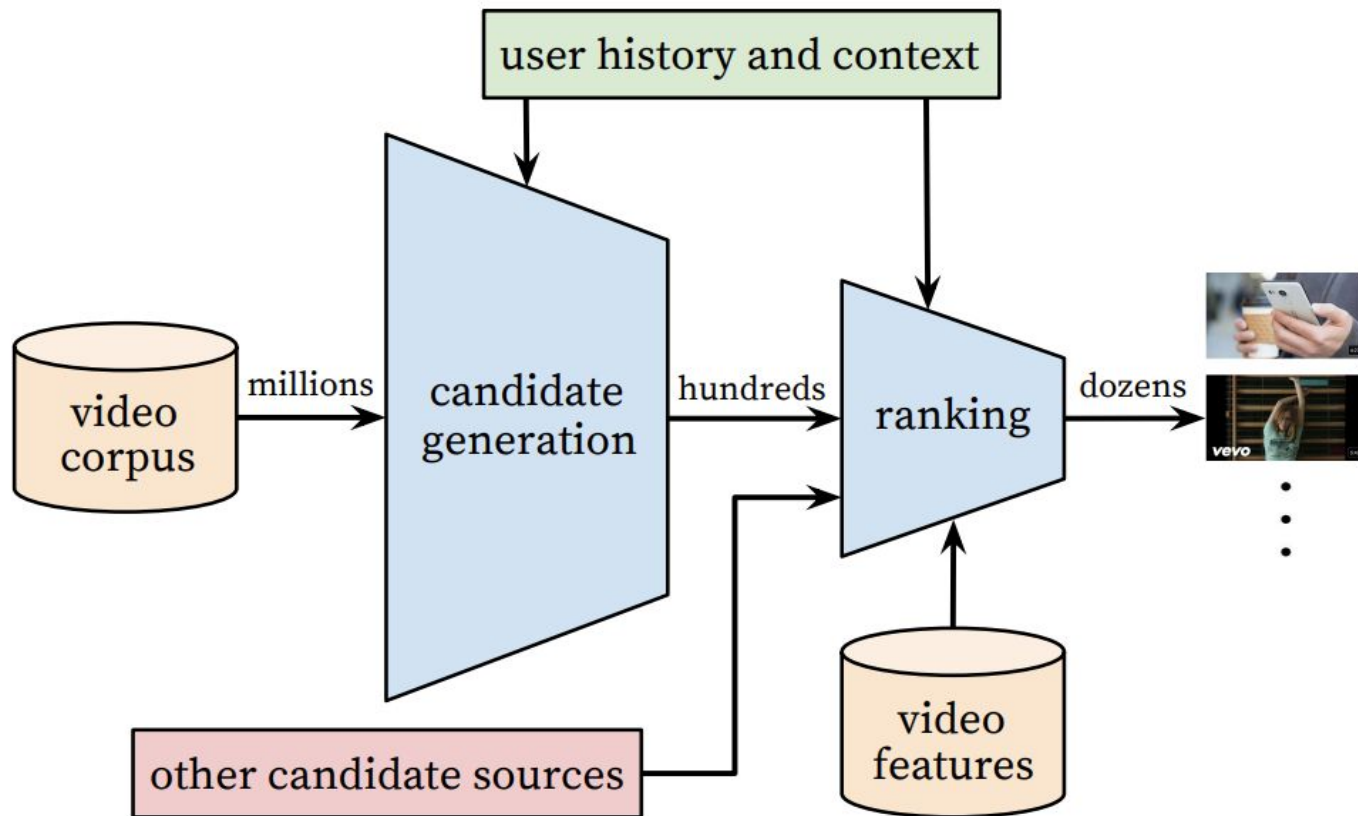# Designing a Recommendation System for the Healthfeed

Four Steps for Building Recommendation system

|	Retrieval	Filtering	Scoring	Ordering
Music Discovery	Find similar songs based on nearest neighbour search	Remove tracks users listened before	Predict likelihood that a user will listen to a song	Trade-Off between score, similarity, BPM, etc
Social Media	Find new posts in user's network	Remove posts from blocked and muted users	Predict likelihood that a user will interact with it	Change order that adjust posts are from different authors
Online Store	Find items which are usually co-purchased	Remove items which are out of stock	Predict likelihood that a user will purchase an item	Reorder items based on price points
Streaming Service	Find items based on different rows/shelves/topics	Remove items which are not available for user's country	Predict user's stream time per item	Organize recommendations to fit genre distributions

Candidate Retrieval & Ranking Architecture
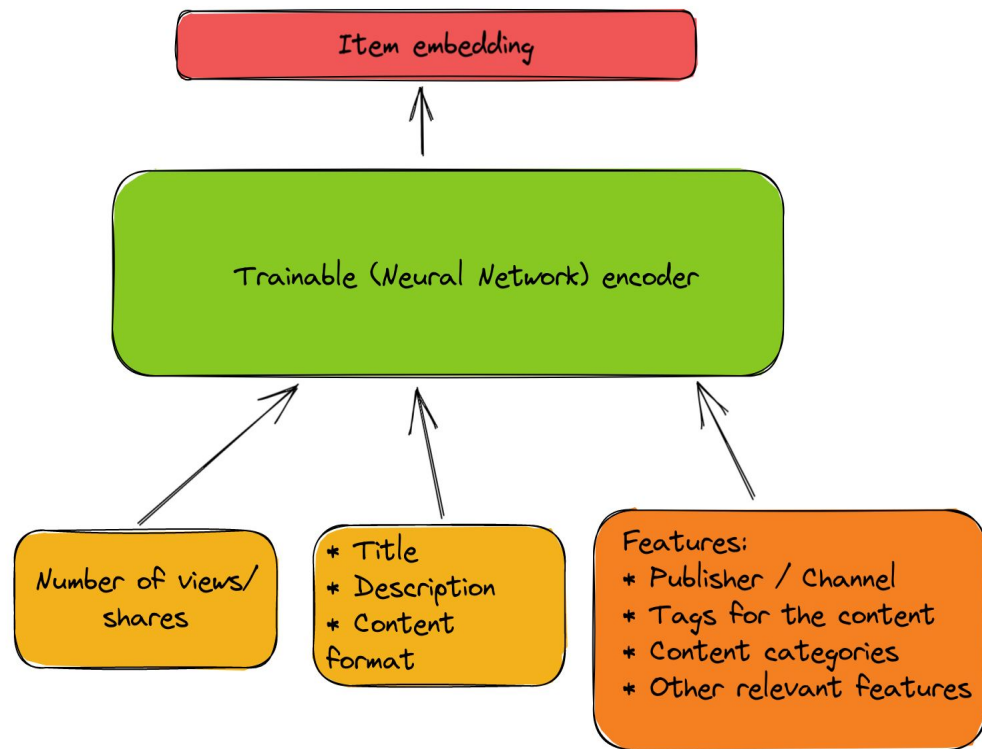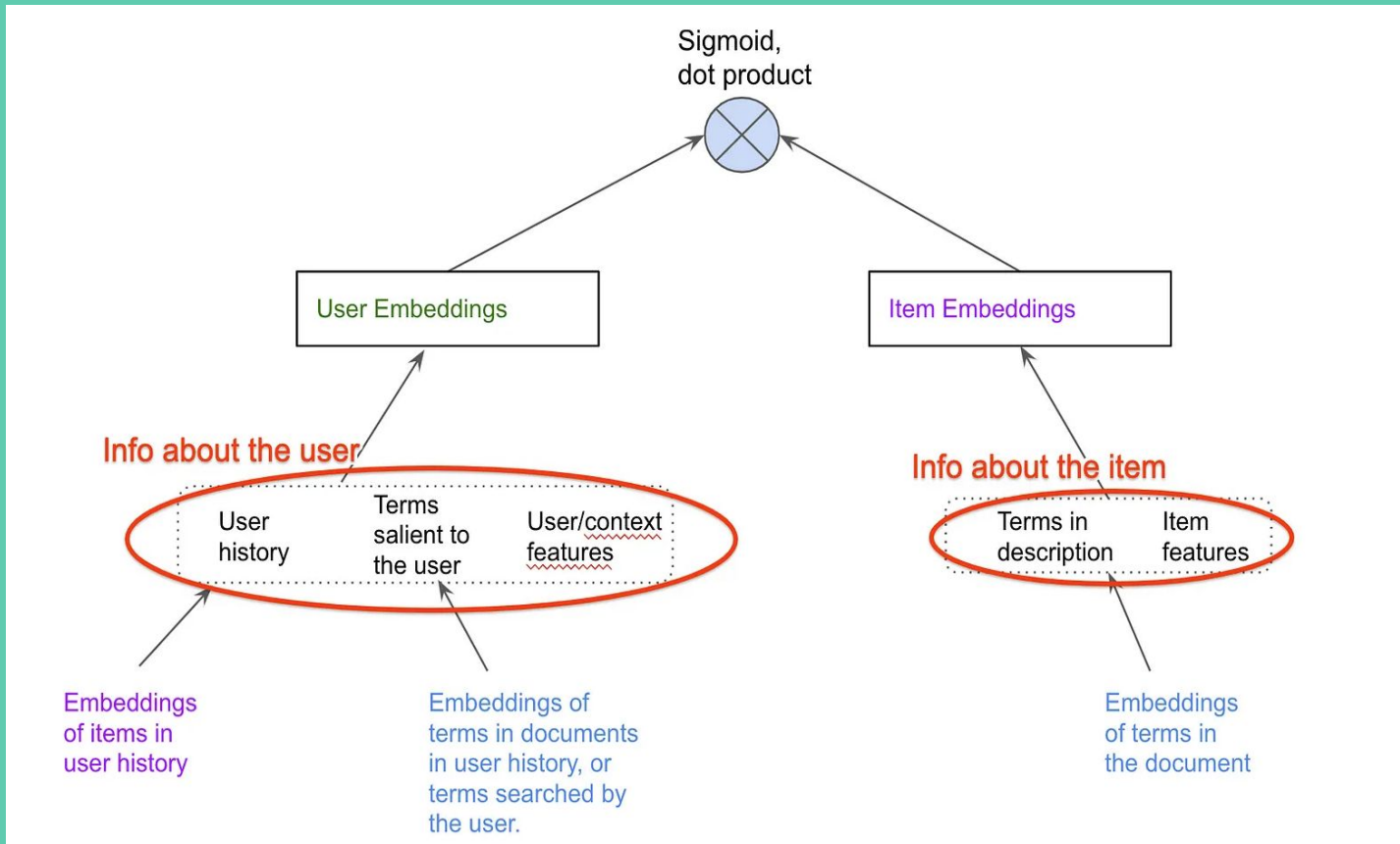
Using Candidate Retrieval and Ranking for Healthfeed
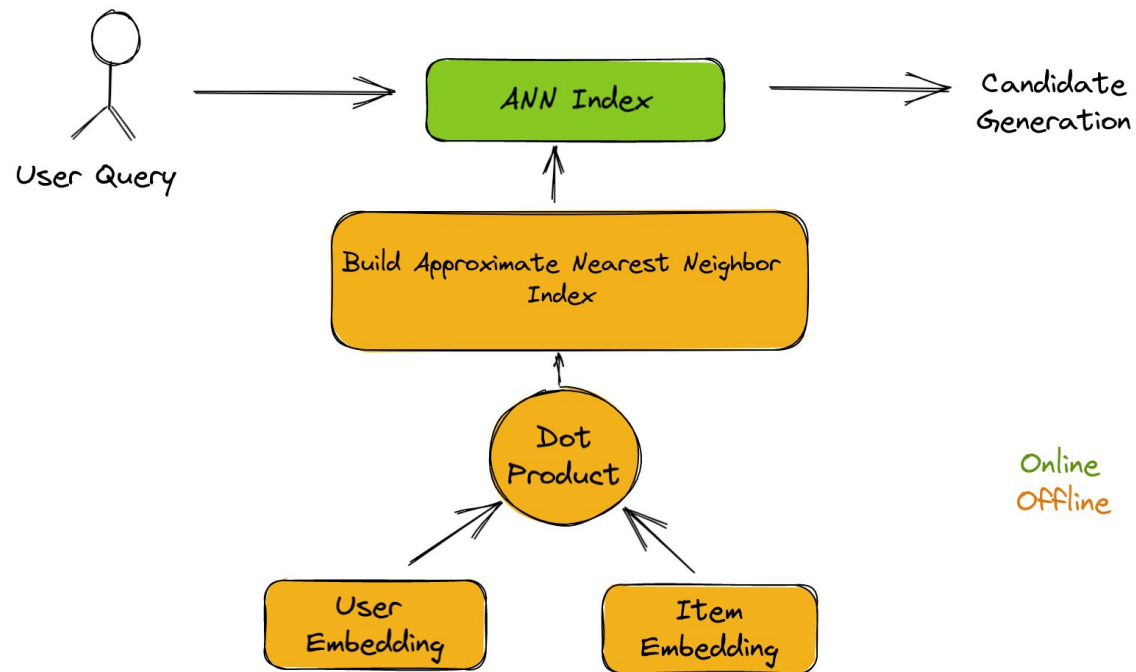
Creating User Embedding for Candidate Retrieval
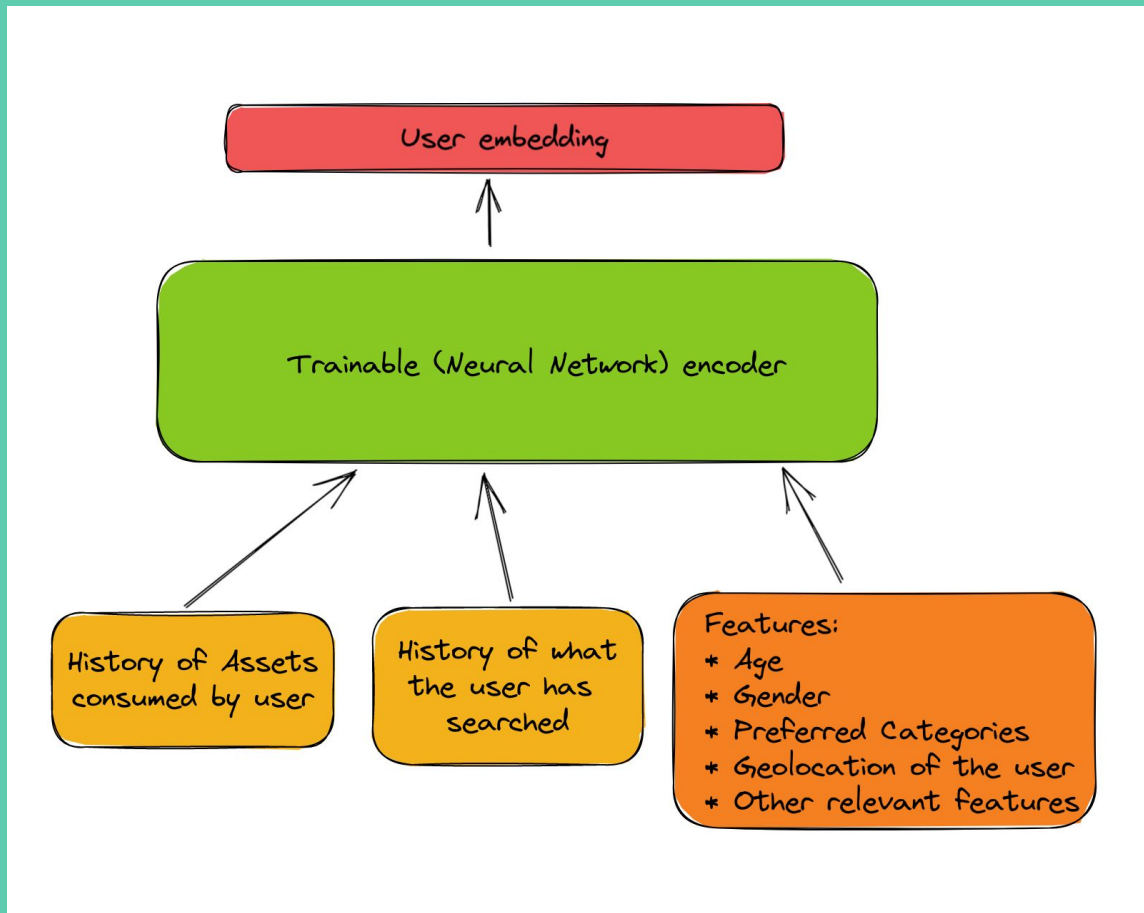
Creating Item Embedding for Candidate Retrieval

Combining User and Item Embedding for Candidate Retrieval

Sigmoid, dot product

User Embeddings

Item Embeddings

Info about the user

User history

Terms salient to the user

User/context features

Info about the item

Terms in description

Item features

Embeddings of items in user history

Embeddings of terms in documents in user history, or terms searched by the user.

Embeddings of terms in the document

Candidate Retrieval from Millions to Hundreds

Candidates Generated → Machine learning Model → Sort the probabilities → Rerank as per Business rules → Show Recommendations

Consider additional User and Item features, Examples
* Freshness of content
* Article word length
* Length of the video
* Time between user last and current login
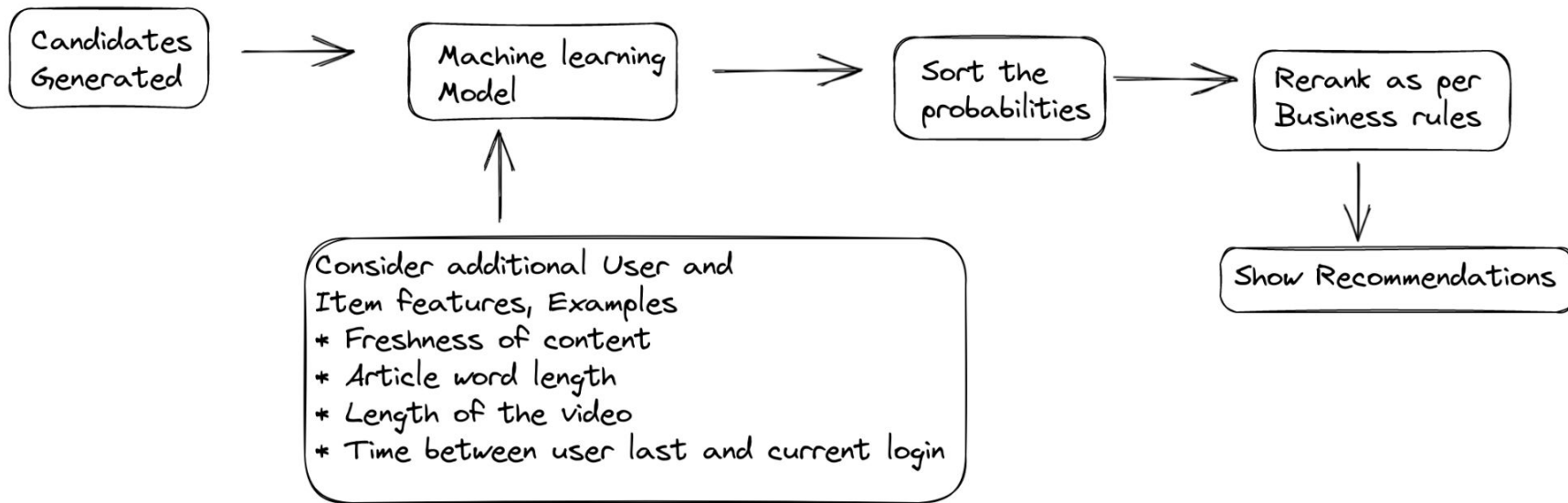
Complete Recommendation Architecture
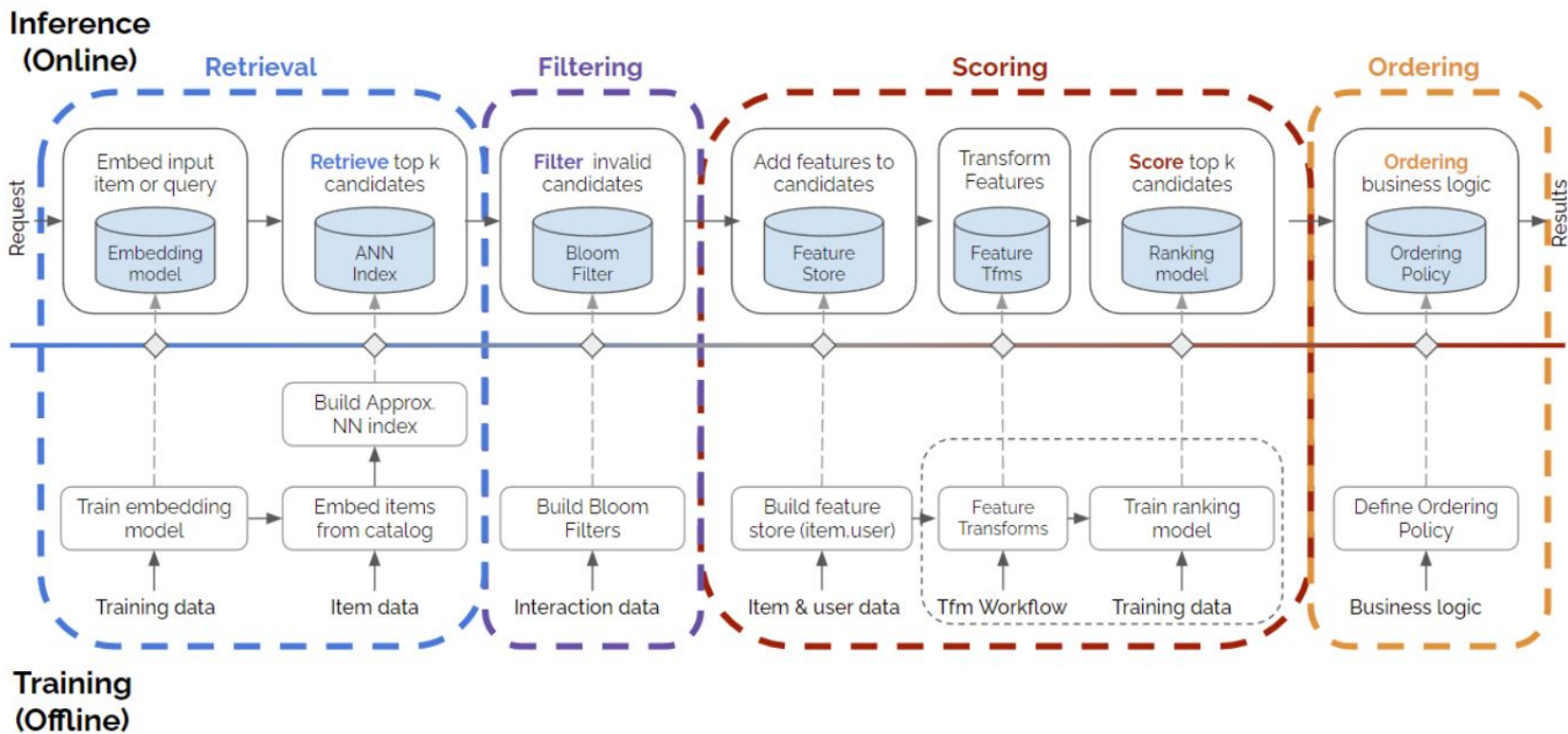
Evaluating Recommendation Systems

Precision@k
Recall@k
Coverage
Novelty
Personalization   →   ML
Intra-list similarity   metrics   →   Business
ROC or AUC   metrics   →   Increase in number of views
MSE or RMSE   Increase in Watch time
                                                          Increase in number of shares
                                                          Ratio of Doctor bookings or lab tests to the
                                                          time spent on Healthfeed

Conclusion

- Design the system based on Business requirements and Data

- Candidate Retrieval and the Ranking system will scale with data

- Start with a simple system and increase the complexity

- AB Testing capability is important for evaluating and improving the recommendation system

- Beware of Feedback loops

r