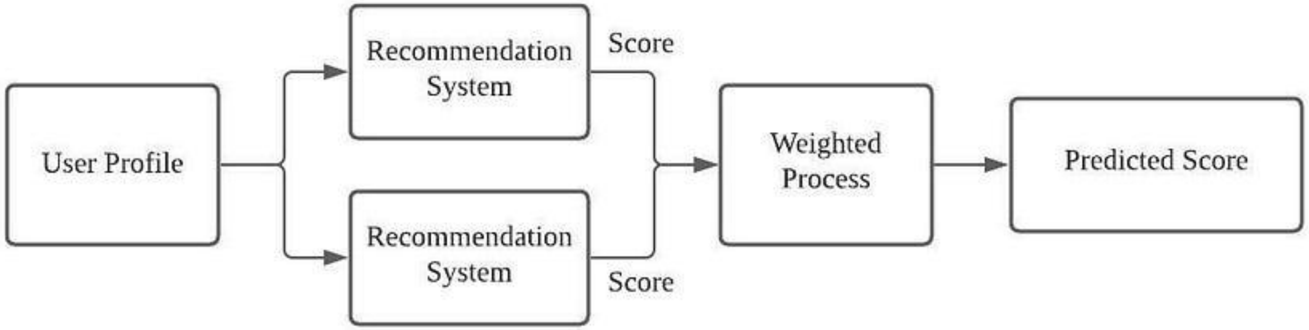
Hybrid Recommenders

Hybrid Recommenders

- Hybrid approaches can be implemented in several ways:
  - by making content-based and collaborative predictions separately and then combining them;
  - by adding content-based capabilities to a collaborative approach (and vice versa);
  - by unifying the approaches into one model.

Seven approaches in building the hybrid recommender system.

1. Weighted

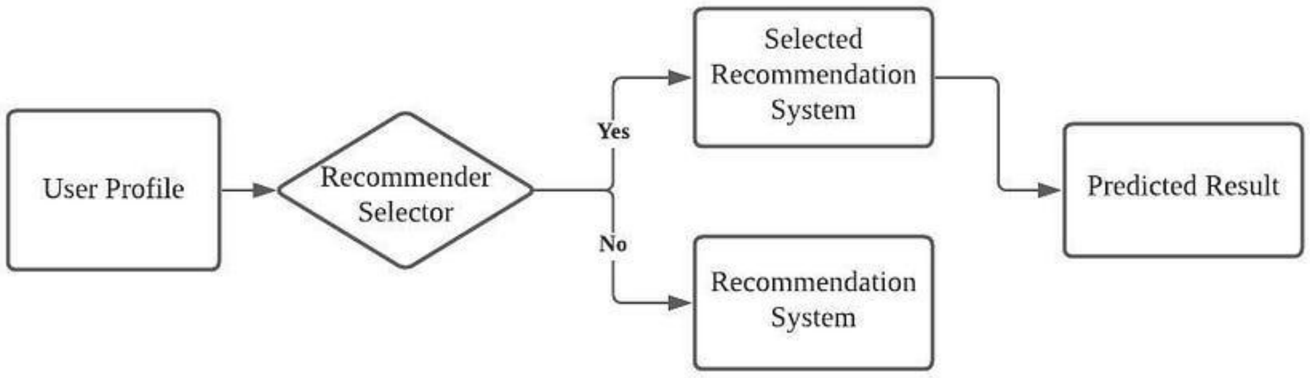


Weighted Hybrid Recommendation System

In the weighted recommendation system, we can define a few models that is able to well interpret the dataset. The weighted recommendation system will take the outputs from each of the models and combine the result in static weightings, which the weight does not change across the train and test set.

For example, we can combine a content-based model and a item-item collaborative filtering model, and each takes a weight of 50% toward the final prediction. The benefit of using the weighted hybrid is that we integrate multiple models to support the dataset on the recommendation process in a linear way.

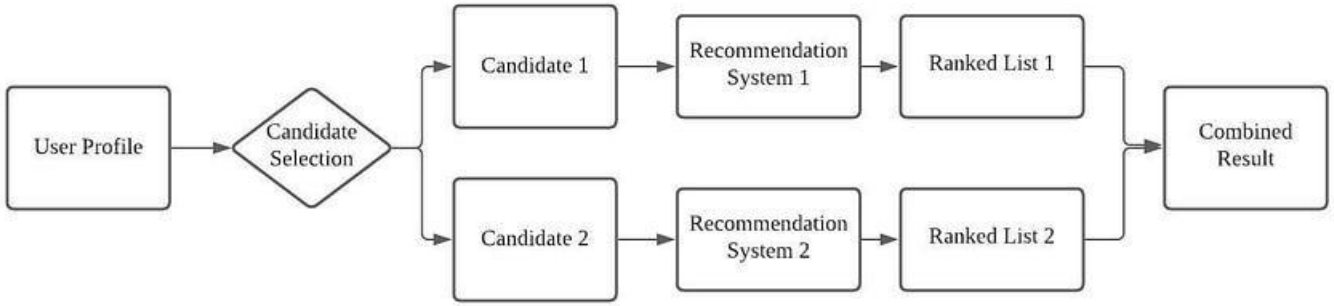
2. Switching



Switching Recommendation System

The switching hybrid selects a single recommendation system based on the situation. The model is used to be built for the item-level sensitive dataset, we should set the recommender selector criteria based on the user profile or other features. The switching hybrid approach introduces an additional layer upon the recommendation model, which select the appropriate model to use. The recommender system is sensitive to the strengths and weakness of the constituent recommendation model.

3. Mixed

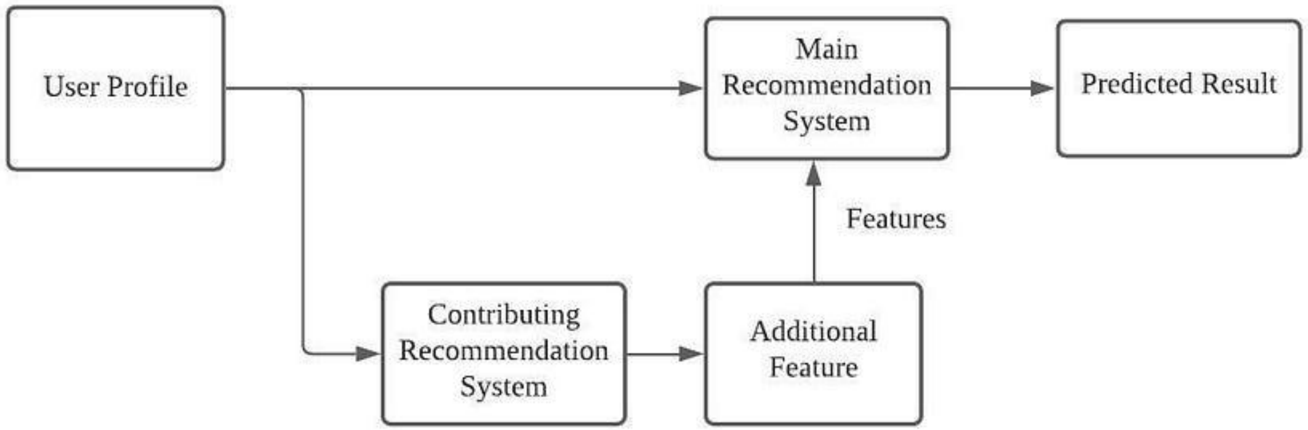


Mixed Recommendation System

Mixed hybrid approach first takes the user profile and features to generate different set of candidate datasets. The recommendation system inputs different set of candidate to the recommendation model accordingly, and combine the prediction to produce the result recommendation.

The mixed hybrid recommendation system is able to make large number of recommendations simultaneously, and fit the partial dataset to the appropriate model in order to have better performance.

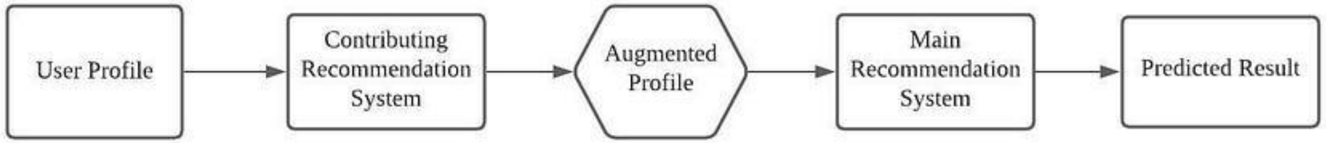
4. Feature Combination



Combination Recommendation System

In feature combination hybrid, We add a virtual contributing recommendation model to the system, which works as feature engineering toward the original user profile dataset. For example, we can inject features of a collaborative recommendation model into an content-based recommendation model. The hybrid model is capable to consider the collaborative data from the sub system with relying on one model exclusively.

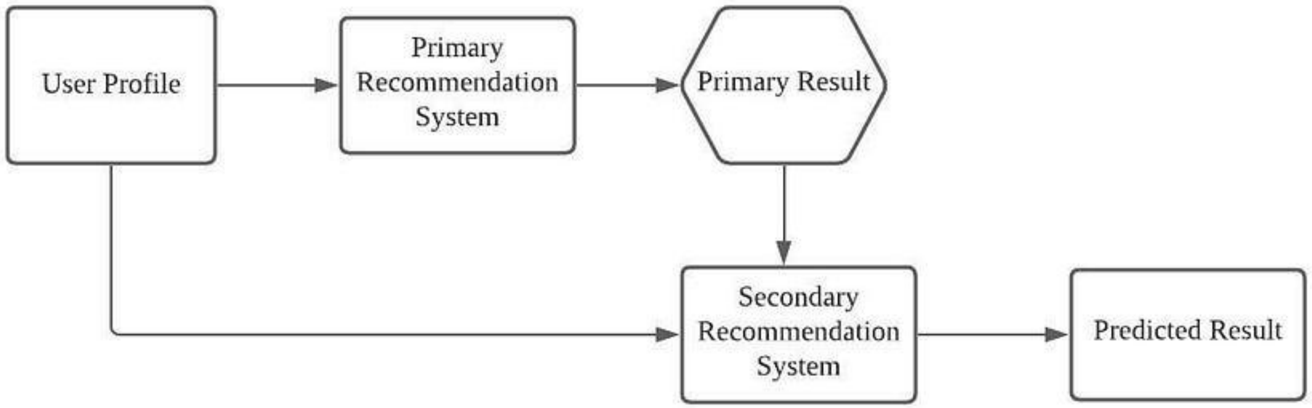
5. Feature Augmentation



Feature Augmentation Recommendation System

A contributing recommendation model is employed to generate a rating or classification of the user/item profile, which is further used in the main recommendation system to produce the final predicted result. The feature augmentation hybrid is able to improve the performance of the core system without changing the main recommendation model. For example, by using the association rule, we are able to enhance the user profile dataset. With the augmented dataset, the performance of content-based recommendation model will be improved.

6. Cascade



Cascade Recommendation System

Cascade hybrid defines a strict hierarchical structure recommendation system, such that the main recommendation system produce the primary result, and we use the secondary model to resolve some minor issues of the primary result, like breaking tie in the scoring. In practice, most of the dataset are sparse, the secondary recommendation model can be effective against equal scoring issue or missing data issue.

7. Meta-Level

Meta-level hybrid is similar to the feature augmentation hybrid, such that the contributing model is providing augmented dataset to the main recommendation model. Different from the feature augmentation hybrid, meta-level replaces the original dataset with a learned model from the contributing model as the input to the main recommendation model.

Benefits of using Hybrid Recommender Systems

- Increased accuracy: Hybrid Recommender Systems can provide more accurate recommendations by combining the strengths of different recommender systems.
- Increased diversity: Hybrid Recommender Systems can provide more diverse recommendations by combining recommendations from different systems.
- Robustness: Hybrid Recommender Systems can be more robust to cold-start problems, where there is not enough data on new users or new items.

EVALUATING THE RECOMMENDATION SYSTEM

https://neptune.ai/blog/recommender-systems-metrics

Evaluating Recommender System

- Mean Absolute Error (MAE)
- Root Mean Square Error (RMSE)
- Hit Rate
- Average Reciprocal Hit Rate (ARHR)
- Coverage
- Diversity
- Novelty

Mean absolute error (MAE)

$$MAE = (\frac{1}{n})\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Mean Absolute Error

This is the most straightforward metric of evaluation known as Mean absolute error. The above is a fancy equation for evaluating it. It is literally the difference between what user might rate a movie to what our system predicts.

Root mean square error (RMSE)

$$RMSE = \sqrt{\left(\frac{1}{n}\right)\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}$$

Root mean square error

This is another common and perhaps most popular metric of evaluation. One reason is that it penalises you way less when you're close to actual prediction and way more when far from actual prediction compared to MAE.

Hit rate

$$HitRate = \left(\frac{hits}{users}\right)$$

Hit Rate

This is a simple metric. First, you generate a top-N recommendation for a user. If one of the recommendations in a user's top-end recommendations is something they actually rated, you consider that a hit. Since the system actually managed to show the user something that they found interesting enough to watch on their own already, so we’ll consider that a success.

So to calculate we add up all the hits in the top-N recommendations for every user and divide it by every user.

Average reciprocal hit rate (ARHR)

$$ARHR = (\frac{1}{users})\sum_{i=1}^{n}(1/rank_i)$$

The average reciprocal hit rate

This is a variation of hit rate but it accounts for wherein Top-N list your hits appear. So we get more credit for recommending items in the top slot than in bottom slot. This metric more of user-focused. If the user has to scroll down to see a lower item in your Top-N list that it makes sense to penalise recommendation that appears too low in the list as the user has to work to find them.

Coverage

In simple words, its the % of (user, item) pairs that can be predicted or percentage of possible recommendation that the recommender system can provide. For example, Think about the MovieLens data set of movie ratings. It contains ratings for several thousand movies, but there are plenty of movies in existence that it doesn't have ratings for.

Therefore if we are using this data for recommending movies on IMDB which contains several millions of movies the coverage would be quite low.

It's worth noting that coverage can be at odds with accuracy. If you enforce a higher quality threshold on the recommendations you make, then you might improve your accuracy at the expense of coverage.

Diversity

$$Diversity = (1 - S) \ where \ S = avg \ similarity \ between \ recommendation \ pairs$$

Diversity

Think of this metric as to how broad a variety of items your recommender systems is showing to users.

Suppose if you watch James Bond movie. Low diversity would be a recommender system that just would recommend next parts of the James Bond series but doesn't recommend other movies which is not a part of James Bond series but still related to the same genre.

Very high diversity is also not always good. Completely random items have high diversity but those are not very good recommendations. You also need to check diversity alongside some other metric that measures the quality of recommendations as well.

Novelty

The novelty in case of recommender systems refers to how popular are the items that it is recommending. (i.e. mean popularity rank of recommended items)

And again, just recommending random stuff would yield very high novelty scores since the vast majority of items are not top sellers. Although novelty is measurable, what to do with it is in many ways subjective.

There's a concept of user trust in a recommender system. People want to see at least a few familiar items in their recommendations.

EVALUATING RECOMMENDER SYSTEM

- Metrics
  - Accuracy, Decision Support, Rank, others
- Evaluation without users
  - Evaluating offline data
  - Framework for hidden-data evaluation
- Evaluation with users
  - Lab and Field Experiments (A/B Trials)
  - User Surveys, Log Analysis