Playlist Recommender

Himanshu Gupta, Mahesh Avadhanam, Nishant Aditya, Pranay Dhariwal

Introduction

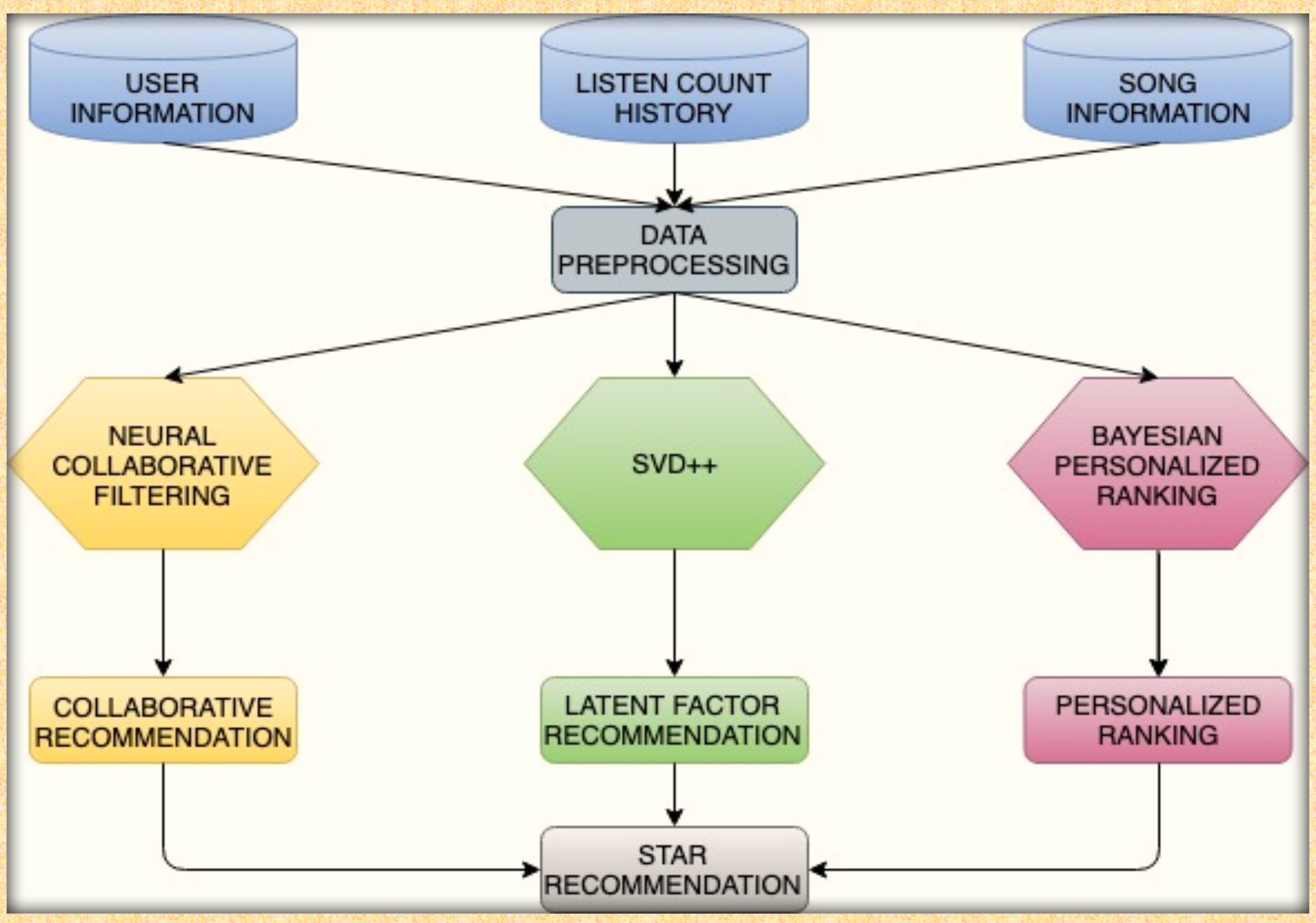
.

- With the presence of internet and online vendors like Spotify, google music, it is important to analyze how these systems work and how they can be improved further.
- We experiment with different approaches and ensemble it to further improve the recommender's performance. Several approaches like SVD, SVD++, Bayesian Personalized Ranking, Neural Collaborative Filtering were implemented to produce a playlist for given user.
- Popularity based model is useful to overcome the cold start problem.

Data Processing

- The Million Songs Data is about 280GB and contains 48M of user-id, song-id, play count obtained through the listening histories of over one million users where play count is treated as implicit feedback.
- We used a subset of above data containing data for 10,000 songs and 2M triplets of 76,353 users
- Used listen count as implicit feedback.
- Scaled down the listen counts by a log factor and used 80% of the data to train and the remaining 20% to evaluate the recommender.

Architecture



The above image explains the overall workflow of our project combining results from different approaches.

Implementation

SVD/SVD++

- Song preferences are influenced by factors specific to domain like genre, artist, etc.,. Users and songs are characterized by these latent factors.
- Used Surprise library to predict the listening scores. SVD++ accounts for implicit feedback of whether user has listened to the song.

Bayesian personalized ranking

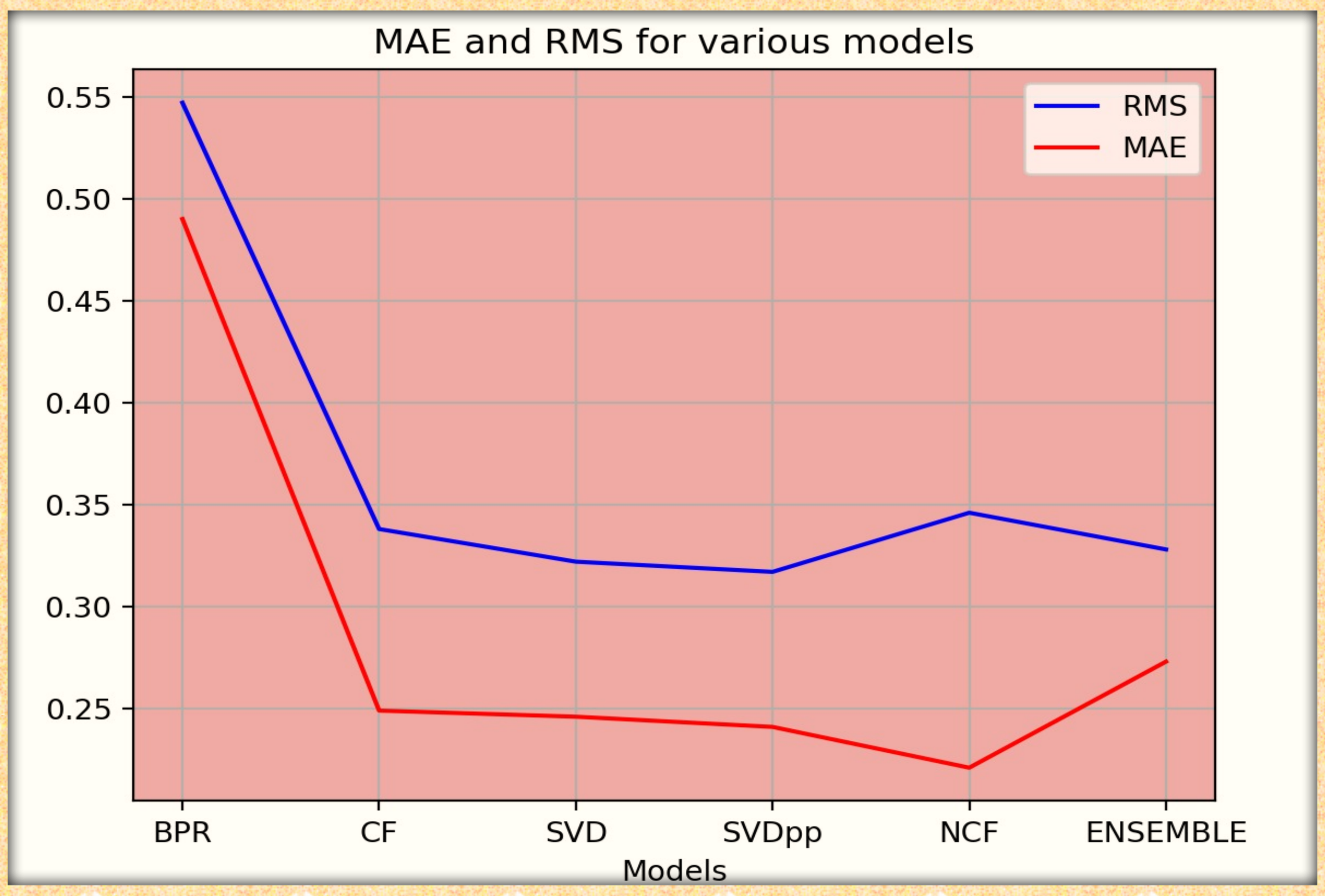
- BPR presents a general optimization technique for personalized ranking using a Bayesian analysis of the problem.
- The users are assumed to be independent of each other and ranking of items among one user is independent from other users.

Neural Collaborative filtering

- We concatenate the user and item to create a feature vector which can be passed to further layers.
- Neural MF can then combine the predictions from Multi layer perceptron and GMF to obtain the following prediction

Collaborative Filtering

- Generating automatic recommendations/predictions about the interests of a user.
- Calculate the expected rating of an item by using the user-to-user similarity and aggregate the scores by considering other users who have also consumed the same item.



Results

Model	RMSE	MAE	Precision@5	NDCG@5
Popularity	N/A	N/A	0.023	0.0339
SVD	0.3225	0.2468	0.8551	0.8384
SVD++	0.3172	0.2418	0.8575	0.8395
CF	0.3388	0.2495	0.8468	0.8311
BPR	0.5473	0.4925	0.8501	0.8357
NCF	0.3461	0.2218	0.8568	0.8392
Ensemble	0.3285	0.2731	0.8585	0.8397

Key Takeaways

- Using an ensemble approach helps us to incorporate the benefits of all the models and overcome the demerits of individual methods.
- BPR model only considers the usage of a particular user and does not accounts for their actual scores which is why it gives higher error values.
- Using metadata and content based recommendation is more practical but challenging to implement.
- Neural collaborative filtering has the least mean absolute error

Future Work

- The combination of different methods can have a weighted approach which can be learnt according to the data.
- Implement the recommender on a distributed system to run parallel computations on the complete dataset. This can help reduce the runtime of current implementations.
- Use content based recommendation which utilizes the several different aspects of the metadata associated with a song like album, singer, genre and release year.