Product Recommendation - PoC

November 6, 2019

The power to xelerate

Change history

Doc Version	Author & Created Date	Change Description & Section	Reviewed By & Review Date	Approved By & Approved Date
1.0	Vaisakh B 06 November 2019	All		

Contents

Product Recommendation Systems

"A system to assist user discover new and relevant products, creating a delightful user experience while driving incremental revenue through cross sell and up sell."

Classification of Recommendation Systems

- Popularity Based Filtering
- Content Based Filtering
- Collaborative Filtering

Popularity Based Filtering

- We can recommend items to a user which are most popular among all the users
- **Non - Personalized Recommendation**

SunTec™

Content Based Filtering

- Serves recommendations based on the meta-data or characteristics of the product
- **Semi – Personalized Recommendation**
- Challenges
  - Lack of novelty and diversity
  - Scalability is a challenge

Collaborative Filtering

- Serves recommendations based on user similarity
- **Personalized Recommendation**

Advantages of Collaborative Filtering

- Benefits from large user bases
- Produces more serendipitous recommendations
- Flexible across different domains

Implementation Methods

- Memory based approach
- Model based approach

Memory Based Approach

- User - User filtering
  - Finds the similarity between users based past rating
  - Predicts the user preference for an item as the weighted sum of user similarities and rating of the given item by different users
- Item – Item filtering
  - Finds the similarity between items based on its rating
  - Predicts the user preference for an item as the weighted sum of item similarities and the given user's rating for different items
- Algorithms
  - KNN
  - Cosine similarity
  - Pearson correlation

Disadvantages of Memory Based Approach

- Scalability
  - When there is large number of users and products, computation power becomes an issue
- Data Sparsity
  - There may be large number of users and products. But user rating for products won't be available in good numbers, in which case recommendation won't be accurate.

Model Based Approach [Latent Factor Method]

- Solves scalability and sparsity problems
- In this approach, CF models are developed using parametric machine learning algorithms to predict user's rating of unrated items
- The idea behind such models is that preferences of user can be determined by a small number of hidden factors
- These factors are called embeddings/latent features

Different Latent Factor Methods

- Matrix Factorization Method
  - We decompose our original user-item rating matrix into product of 2 low rank orthogonal matrices, which represents the embeddings
  - We will be using Funk SVD method (Regularized SVD)
- Deep Learning
  - Hidden layers models embeddings / new feature space

Evaluation of Recommendation Systems

- User Studies
- Online Evaluation
- Offline Evaluation

Major Challenges

- Cold Start Problem
  - Hybrid approach
    - Product Cold Start => Content based filtering
    - Visitor Cold Start => Demographic clustering + Popularity based strategy
- Data
  - Data Collection
    - Explicit
    - Implicit
      - Rating function based on behavioral data
      - Time decay algorithm
      - Inverse frequency factor
- Anonymous Users

State of the Art

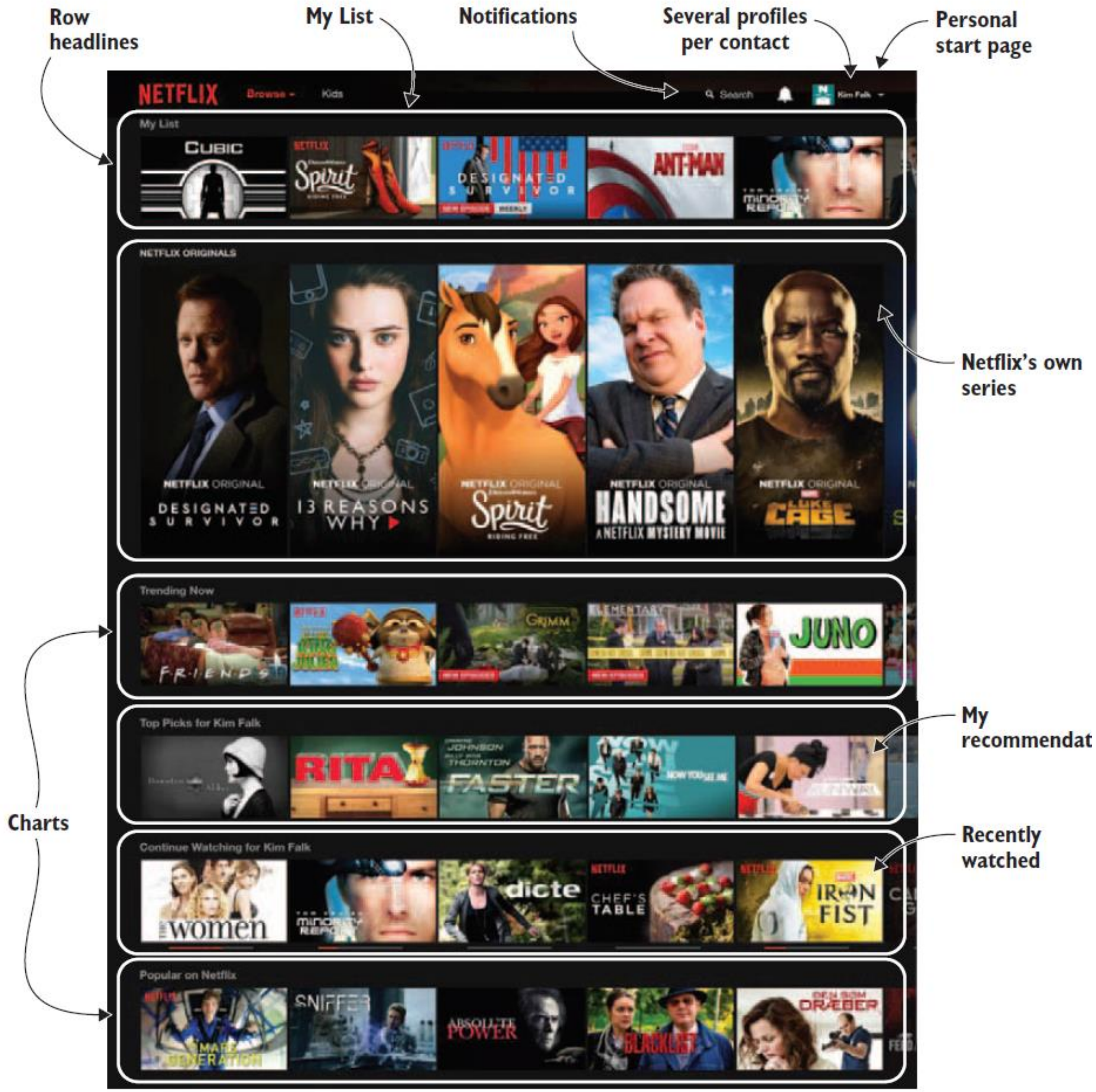

Figure 1.1 The Netflix start page (before it changed the layout)

Appendix 1 -
CF Memory Based Method Prediction Equation

- User – User filtering

$$P_{u,i} = \frac{\sum_v (r_{v,i} * s_{u,v})}{\sum_v s_{u,v}}$$

- Item – Item filtering

$$P_{u,i} = \frac{\sum_N (s_{i,N} * R_{u,N})}{\sum_N (|s_{i,N}|)}$$

SunTec™

Appendix 2 - Algorithm : Funk SVD Method

- Define baseline prediction function
- Calculate the error function
- Optimize the error function using SGD

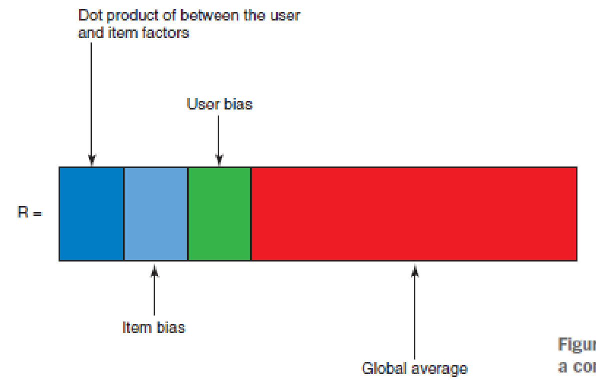

Figure 11.16 A predicted rating is a combination of these four things.

$$\min_{b,p,q} \sum_{(u,i)\in K} \left(r_{iu} - \mu - b_u - b_i - q_i p_u\right)^2$$

- $b_u \leftarrow b_u + \gamma * (e_{ui} - \lambda * b_u)$
- $b_i \leftarrow b_i + \gamma * (e_{ui} - \lambda * b_i)$
- $q_i \leftarrow q_i + \gamma * (e_{ui} * p_u - \lambda * q_i)$
- $p_u \leftarrow p_u + \gamma * (e_{ui} * q_i - \lambda * p_u)$

References

- Practical Recommender Systems Book by Kim Falk
- https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-recommendation-engine-python/?

SunTec™

Thank You

The power to xelerate