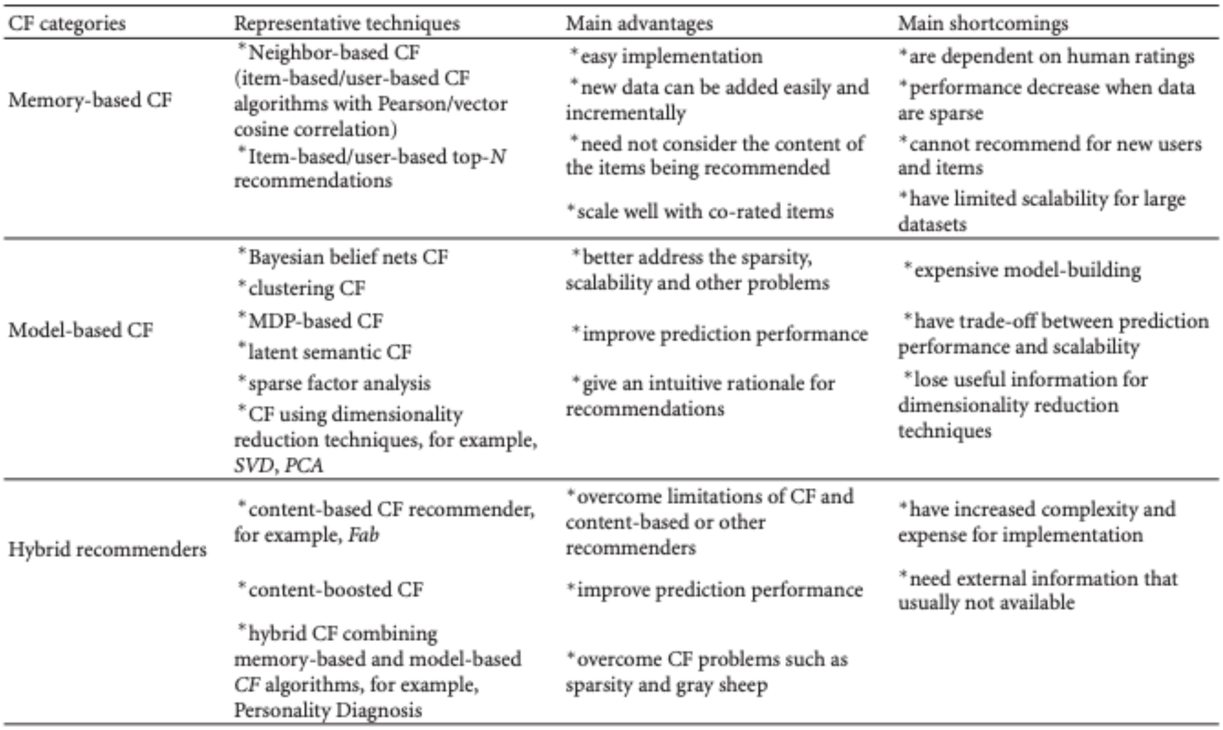
Section 3

What kind of algorithms would you explore to solve this issue? What's your preferred model? How would you compare different models, and why? Explain the pros and cons of each of these models.

I would use Collaborative filtering based algorithm to solve this issue. My preferred model that would use dimentionality reduction based technique such as matrix factorization and SVD. To compare different models best option is to do research find one and two popular models and quickly implement them to see performances. A survey paper like A Survey of Collaborative Filtering Techniques by Xiaoyuan Su and Taghi M. Khoshgoftaar (https://www.hindawi.com/journals/aai/2009/421425/) can give us a good overview of various algorithms. Here is pros and cons various Collaborative filtering based algorithm collected from the same paper https://www.hindawi.com/journals/aai/2009/421425/tab2/ (https://www.hindawi.com/journals/aai/2009/421425/tab2/)

Table 2: Overview of collaborative filtering techniques.



CF categories	Representative techniques	Main advantages	Main shortcomings
Memory-based CF	*Neighbor-based CF (item-based/user-based CF algorithms with Pearson/vector cosine correlation) *Item-based/user-based top-*N* recommendations	*easy implementation *new data can be added easily and incrementally *need not consider the content of the items being recommended *scale well with co-rated items	*are dependent on human ratings *performance decrease when data are sparse *cannot recommend for new users and items *have limited scalability for large datasets
Model-based CF	*Bayesian belief nets CF *clustering CF *MDP-based CF *latent semantic CF *sparse factor analysis *CF using dimensionality reduction techniques, for example, *SVD, PCA*	*better address the sparsity, scalability and other problems *improve prediction performance *give an intuitive rationale for recommendations	*expensive model-building *have trade-off between prediction performance and scalability *lose useful information for dimensionality reduction techniques
Hybrid recommenders	*content-based CF recommender, for example, *Fab* *content-boosted CF *hybrid CF combining memory-based and model-based *CF* algorithms, for example, Personality Diagnosis	*overcome limitations of CF and content-based or other recommenders *improve prediction performance *overcome CF problems such as sparsity and gray sheep	*have increased complexity and expense for implementation *need external information that usually not available

Let's assume we have chosen to work with a matrix factorization model.

- **What are the steps/techniques you use to make sure that you're not over-fitting your model?**
    - I would use a regularization parameters to avoid over-fitting my model

- **What techniques would you use to detect outliers?**
    - To detect outliers I would run an clustering algorithm such as PAM (Partition around medoid). I also wrote myself a detail tutorial how to implement it using R. You can access it here (https://medium.com/@rumman1988/clustering-categorical-and-numerical-datatype-using-gower-distance-ab89b3aa90d9)

- **How would you solve the cold start problem? (i.e. how would you update the algorithm so it not only can make recommendations to the existing users in the recommender, but also to new users)**
    - For new users we can build their persona using the following techniques recommend based on their persona
        - Based on new user's details such as age, demographic, location etc find similar existing user to him
        - Track users initial shopping habit
        - Use available external data such as data from different sources, social media etc

- **Propose a model that incorporates both the ratings and the Shoppers and Products attributes (age, gender, location for shoppers, type, size, color for products)? Describe your technique.**
    - From age, gender, location for shoppers we can classify users of similar persona
    - For type, size, color of products we can classify products of similar attributes
    - From this we can use content based recommendation based on user's previous ratings
    - Now the build our algorithm based on our hypothesis users of similar personas will like product of similar attributes
    - In matrix factorization we determine user of similar personas by comparing similarity of ratings of same products amongst users. In this case we can use this additional attributes as well. This would be a hybrid approach of collaborative filtering and content based recommendation
    - When we decide whether to recommend a product to a particular user or not we can consider whether users of similar persona liked products of similar attributes or not.

Assume we have 1 instance of our model per store (each store has its own recommender) due to resource (memory and time) limitations. How would you efficiently recommend items from one store to another? These stores can share shoppers. Describe your solution(s).

- Assuming we have Shoppers and Product attributes we can use content based recommendation across the stores.For example if user highly rated some items in a previously visited store we can recommend items similar to that from his current stores.
- For two similar users if one user rated highly one item in one store, other user can be recommended a similar item from a different store

Resource used

- https://www.coursera.org/lecture/machine-learning-with-python/content-based-recommender-systems-jPrfc (https://www.coursera.org/lecture/machine-learning-with-python/content-based-recommender-systems-jPrfc)
- https://engineering.shopify.com/blogs/engineering/how-shopify-uses-recommender-systems-to-empower-entrepreneurs (https://engineering.shopify.com/blogs/engineering/how-shopify-uses-recommender-systems-to-empower-entrepreneurs)
- https://en.wikipedia.org/wiki/Collaborative_filtering (https://en.wikipedia.org/wiki/Collaborative_filtering)
- https://www.hindawi.com/journals/aai/2009/421425/ (https://www.hindawi.com/journals/aai/2009/421425/)
- https://towardsdatascience.com/paper-summary-matrix-factorization-techniques-for-recommender-systems-82d1a7ace74 (https://towardsdatascience.com/paper-summary-matrix-factorization-techniques-for-recommender-systems-82d1a7ace74)

```
In [ ]:
```