Below I've listed a number of ML Algorithms we could use with a short description and overview of strengths, weaknesses, and requirements. It is noteworthy that hybrid models exist between all of these methods.

1. Collaborative Filtering

Collaborative filtering relies on user-product interactions like ratings or purchases to make recommendations. It assumes that if users A and B have similar behavior, they will like the same products.

- Strengths:
  - Quite personalized recommendations.
  - based on user behavior alone.
  - Works well with a large user base and lots of product interactions.
- Weaknesses:
  - Struggles with new users or products without sufficient interaction data.
  - Requires significant amounts of user interaction data to perform well.
  - Works well in later stages of the project, once sufficient user-product interaction data has been collected but not the best when dataset is small
- Data Requirements: User-product interaction data such as ratings, purchases, or product engagement. This will likely not be a good fit for the purposes of this class but a good idea overall.

2. Matrix Factorization

Matrix factorization is a type of collaborative filtering that decomposes the user-product interaction matrix into latent factors, which helps to identify underlying patterns between users and products. Basically, Users who like certain skincare products may share latent preferences for products containing similar ingredients (e.g. products with hyaluronic acid).

- Strengths:
  - Works well with large datasets.
  - Efficient at uncovering hidden patterns in user-product interaction data.
- Weaknesses:
  - Like collaborative filtering, it faces a slow start problem
  - Requires larger datasets to be effective, having little interaction data can lead to poor performance.
- Data Requirements: Requires a user-product interaction matrix with sufficient data to factorize.

3. Content-Based Filtering

Content-based filtering makes recommendations based on the similarity between products. It uses product features like type, ingredients, price and user profiles like skinType and concerns to recommend items similar to those the user has liked. Example If a user prefers moisturizers with hyaluronic acid for dry skin, the system will recommend similar products based on these attributes.

- Strengths:
  - No cold start issues with new products, as recommendations are based on product attributes.
  - Can work well with small datasets, relying on product metadata instead of user interaction data.
  - Offers immediate value without requiring extensive user-product interactions.
- Weaknesses:
  - Limited personalization, as recommendations focus on product attributes rather than user preferences.
  - May fail to discover new products that a user might like but doesn't match past interactions.
- Data Requirements: Requires detailed data about products and a user profile. This will likely be our best option especially with no interaction data.

5. Neural Collaborative Filtering

Description: This is a deep learning-based approach that extends traditional collaborative filtering by using neural networks to model complex interactions between users and products.

- Strengths:
  - Capable of learning very complex patterns and relationships between users and products.
  - Performs well with larger datasets as deep learning models tend to excel with more data.
- Weaknesses:
  - Requires a significant amount of data and large computational resources to train.
  - Harder to interpret and explain compared to simpler models.
- Data Requirements: A large user-product interaction dataset is essential for effective training. I thought it was necessary to mention but this and similar complex models are not too appropriate for our project.

My overall recommendation is Content-Based Filtering. This will allow the system to start providing recommendations immediately based on product attributes like type, price, and ingredients, even when user interaction data is limited. Also the schema includes skinType and concerns stored as bitwise numbers, these can be easily encoded to match product attributes. In a more complete application we can shift towards hybrid models including things like collaborative filtering or even potentially neural collaborative filtering as a long term model for deeper insights