**Horizon Hobby Product Recommendation System Report**

**1.Objective**

To build a personalized product recommendation system using the Horizon Hobby (HH) dataset. The goal is to provide product suggestions to users based on their transaction history, user and product attributes, and optionally promotional activity.

**2.Data Overview**

The HH dataset follows a star schema structure with one fact table and multiple dimension tables.

* fact\_transactions: Contains transaction\_id, customer\_id, product\_id, quantity, price, transaction\_date
* dim\_customers: customer\_id, name, gender, age group, region, loyalty tier
* dim\_products: product\_id, category, subcategory, brand, base\_price, status
* dim\_dates: full\_date, day\_of\_week, month, quarter, fiscal\_year
* dim\_promotions: promotion\_id, type, start\_date, end\_date, target\_segment
* dim\_promotion\_products: product\_id, promotion\_id
* age\_breakout and gender\_breakout: redundant segment views

**3.Data Preparation**

Merged fact\_transactions with dim\_customers, dim\_products, and dim\_dates to create a unified dataset.

Filtered for purchase transactions only.

Created the following features:

User-level:

* total\_spent\_per\_user
* total\_transactions
* avg\_transaction\_value
* recency (days since last transaction)

Product-level:

* total\_units\_sold
* popularity\_rank
* discount\_ratio (if promotion data used)

Interaction-level:

* was\_product\_on\_promotion
* purchase\_season (e.g., Q1/Q2...)

**4.Baseline Models**

**4.1 Global Popularity Model**

Recommended top-selling products by total quantity sold, irrespective of user.

Implementation:

product\_sales = df.groupby('product\_id')['quantity'].sum().reset\_index()  
top\_products = product\_sales.sort\_values('quantity', ascending=False).head(10)

Evaluation:

* Simple and effective for new users
* No personalization

A screenshot of a computer

AI-generated content may be incorrect.

**4.2 Segmented Popularity Model**

Recommended top-selling products by customer segment (e.g., gender or age group).

Implementation:

grouped = df.groupby(['gender', 'product\_id'])['quantity'].sum().reset\_index()  
ranked = grouped.groupby('gender').apply(lambda x: x.sort\_values('quantity', ascending=False).head(10))

Evaluation:

* Improves personalization for cold-start users
* Still static and non-personal beyond segment

**5.Collaborative Filtering Model (ALS)**

Approach:

* Created user-product interaction matrix (implicit feedback: quantity)
* Trained matrix factorization model using ALS (Alternating Least Squares)

Libraries: implicit / LightFM

Implementation:

from implicit.als import AlternatingLeastSquares  
model = AlternatingLeastSquares()  
model.fit(interaction\_matrix)

Evaluation:

* Metrics: Precision@K, Recall@K, NDCG
* Split dataset temporally (e.g., train on past months, validate on recent)

A screenshot of a computer

AI-generated content may be incorrect.

**6.Content-Based Model**

Used product metadata (category, brand, price) and customer demographics (age group, gender) to compute similarity.

Approach:

* Created vector representations of products and users
* Used cosine similarity to match users with similar products

Used TF-IDF encoding + cosine similarity OR product embeddings

Evaluation:

* Works for new products or users with profile data
* Not great for implicit preferences without user behavior

A screenshot of a computer

AI-generated content may be incorrect.

**7.Hybrid Recommendation**

Combined ALS collaborative scores + content-based scores using weighted average.

Also integrated promotional history and seasonality (e.g., users tend to buy X during summer promotions).

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**8.Evaluation Strategy**

* Offline validation using temporal holdout (e.g., last 4 weeks)
* Metrics: Precision@5, Recall@5, NDCG@5
* Analyzed performance across:
* Heavy vs. light users
* Product categories
* Age groups

**9.Results Summary**

| **Model** | **Precision@5** | **Recall@5** | **NDCG@5** |
| --- | --- | --- | --- |
| Global Popularity | 0.035 | 0.082 | 0.044 |
| Segmented Popularity | 0.045 | 0.095 | 0.052 |
| ALS Collaborative | 0.072 | 0.162 | 0.094 |
| Hybrid Model | 0.084 | 0.177 | 0.108 |

**10.Future Improvements**

* Incorporate time-decay (more recent interactions matter more)
* Add reinforcement learning for continuous feedback loops
* Apply business constraints (e.g., in-stock only, margin optimization)
* Real-time serving using a REST API

**11.Conclusion**

The project demonstrated a practical end-to-end recommender system architecture using real-world transactional data from Horizon Hobby. Starting with baselines, we progressively introduced personalization using collaborative and content-based techniques. Final hybrid models significantly outperformed baselines and offer strong potential for operational deployment.