DataQuest 2025 Recommender Systems Challenge

Author: *Lesley Ngcobo*
Date: *2025*

1. Project Overview

This project was developed in response to the **DataQuest 2025 Recommender Systems Challenge** hosted by Bernard Spies of nav.D2–Personalized Solutions. The objective was to build a recommender system that delivers relevant, personalized offers to customers using contextual retail data.

2. Problem Statement

The delivery of personalized offers should be prioritized and ranked based on customer needs derived through contextual data points. To deliver the best possible customer experience, a recommender system model must be developed that recommends the most relevant offers to each customer.

3. Challenge Requirements

1. Train a recommender system model using the provided retail dataset.
2. Evaluate the model using accuracy and beyond-accuracy metrics.
3. Describe considerations for live production use.

4. Tools & Technologies

- **R & RStudio:** Main data cleaning, modeling, evaluation, visualization
- **Excel:** Used for visualizing intermediate results
- **PowerPoint:** For final presentation and storytelling
- **Git & GitHub:** Version control and portfolio publishing

5. Dataset

The dataset includes retail transactions with the following key columns:

- `InvoiceNo`, `StockCode`, `Description`, `Quantity`, `InvoiceDate`, `UnitPrice`, `CustomerID`, `Country`

A cleaned version was created by:

- Removing missing `CustomerIDs`
- Excluding negative quantities and zero/negative prices

6. Recommender Systems Implemented

6.1 Popularity-Based Recommender

- Recommended products based on global purchase frequency
- Also implemented **group-specific popularity** based on Country and Month (seasonality)

6.2 User-Based Collaborative Filtering (UBCF)

- Used cosine similarity to recommend items purchased by similar users
- Found to underperform due to sparsity

6.3 Item-Based Collaborative Filtering (IBCF)

- Recommended items similar to those a user has interacted with
- Achieved strong performance compared to UBCF

6.4 Hybrid Recommender

- Combined UBCF + IBCF using:
  - Equal weighting (score averaging)
  - Tuned weighting (70% IBCF, 30% UBCF)

6.5 Final Hybrid: 4-Signal Model

- Combined:
  - IBCF
  - UBCF
  - Country-based popularity
  - Month-based (seasonal) popularity
- Used a weighted voting strategy for top-N recommendation

7. Evaluation Metrics

Used ***Precision@5*** and ***Recall@5*** to evaluate top-N recommendation performance:

MODEL	PRECISION@5	RECALL@5
USER-BASED CF (UBCF)	0.0014	0.0071
ITEM-BASED CF (IBCF)	0.0167	0.0835
HYBRID (UBCF + IBCF, 70/30)	0.0167	0.0835
HYBRID + GROUP POPULARITY	**0.0181**	**0.0906**

8. Key Insights

- ***UBCF*** underperformed due to sparse data and limited overlap between users.
- ***IBCF*** performed better due to stronger item co-purchase patterns.
- ***Hybrid models*** improved accuracy by combining collaborative perspectives.
- ***Adding country and seasonal popularity*** helped contextualize recommendations, leading to the best performance overall.

9. Production Considerations

If deployed in a live environment:

- Cold-start users should be addressed with ***popularity or content-based*** signals.
- Recommendations should be ***updated regularly*** based on new transactions.
- Performance should be tracked with ***A/B testing*** and real user engagement metrics.
- Model training must be ***efficient and scalable*** on large datasets.

10. Next Steps

- Add matrix factorization and model stacking using Python (e.g., `Surprise`, `LightFM`, or ensemble learning)
- Visualize model comparisons using R or Excel
- Prepare and present findings using slides and performance charts

11. References

- recommenderlab R package
- UCI Online Retail II Dataset
- DataQuest 2025 Challenge prompt (Bernard Spies, nav.D2–Personalised Solutions

12. Acknowledgements

I would like to acknowledge the use of ***OpenAI's ChatGPT*** as a critical learning and support tool throughout this project.

ChatGPT assisted in:

- Planning the recommender system structure and progression
- Debugging and refining R code
- Suggesting evaluation strategies and interpretation of metrics
- Generating technical documentation, README structure, and this report
- Providing clarity and guidance through explanation of recommender logic

All experimentation and implementation decisions were made independently. This project reflects my personal development and my responsible use of AI tools to support that growth.