Project Name:
Ecommerce Recommendation System

Business Objective:

- The goal of this project is to build a recommendation system for an e-commerce platform so users see items they are more likely to engage with.
- The system should help users discover relevant products by learning from their past behavior (personalized recommendation) and from item information.
- Handle the sparse data problem (most users interact with very few items).
- Improve coverage (recommend even for new or inactive users).

Success Criteria:
A personalized recommendation system that performs better than a popularity based system and makes accommodations for new users.

Data Understanding

Data Sources:

- Sourced from an anonymous ecommerce website
- 4 csv files of events (user behavior) , item properties (metadata) and category hierarchy

Data Overview:

- The user–item interaction matrix is very sparse (most users only interact with a small set of items).
- Some items appear in only a few events, making them hard to recommend.

Assumptions/Limitations:

- A lot of identifying data is hashed for anonymity so would affect interpretability of analysis and model deployment.

Data Preparation

Data Cleaning Actions Taken:

- Timestamps converted to readable version
- Datatypes converted to the appropriate datatypes
- Item property table which is in vertical form converted into horizontal form.
- Horizontal Item property table sorted by timestamp and older entries dropped so only most current properties are maintained.

- Hierarchy created in category tree table to establish proper relationships between categories.
- In the horizontal item property table, dropped columns with 95% of entries being null, then merged with the category tree table.
- Merged the resulting table with the event table creating a final table.
- Assigned weights to various events depending on impact.
- Filtered the final table on only available items so the model will not be trained with items no longer available.
- Created an interaction table with the sum of the event weights for each visitor-item pair for implicit modeling.
- Capped outlier interaction scores in the 99.9th percentile using winsorization.
- Encoding: Converted item IDs and user IDs into numeric indices for modeling.
- Train–test split: Used chronological split – train on earlier interactions, test on later ones.
- Matrix building: Built a user–item sparse matrix for the ALS model.
- Weighting: Applied BM25 weighting to reduce popularity bias and emphasize meaningful interactions.
- Category hierarchy expansion: Added ancestor categories for items to improve metadata similarity.

Modeling

Modeling Techniques Tested:

- Popularity based (establishes a baseline) : Always recommends the most popular items.
- ALS : Collaborative filtering
- BPR

Train-Test Split:
80% train, 20% test

Performance Metrics:

- Precision@K
- MAP@K

Selected Model:
ALS with:

- reweighting
- hyperparameter tuning
- hybriding with
  - Popularity (handle cold-start / sparse users)
  - Metadata (coverage and diversity)

Final Hybrid Model = 0.6 (ALS) + 0.3 (Popularity) + 0.1 (meta data similarity)

Evaluation

Model Performance Summary:
The model recommends products correctly with 0.28% precision and 19.43% of the recommendations are relevant to visitors.

Validation Techniques Used:

- Precision@K : Measures how many of the top 10 recommended items were actually relevant.
- MAK@K : Measures ranking quality, rewarding correct items that appear higher up in the list.

Findings

- ALS alone improved personalization compared to popularity, but struggled for very sparse users.
- BM25 weighting boosted Precision@10
- Hybrid model achieved the best overall performance
- Cold starts provided recommendations for new/sparse users.

Deployment

Deployment Strategy:

- Serve the model in e-commerce web application.

Model Monitoring Plan:

- Weekly updates to data and model training, after adding new data
- Track Precision@K and MAP@K over time with A/B testing.
- Update pipeline: Refresh item popularity and metadata similarity regularly.

Appendix

Code Repositories / Scripts:
https://github.com/rolanda4/ecommerce_recommendation_system