Deep Learning-Based Recommendations Using Amazon Purchase Patterns

Ronne Kent Samridhi Singh

Northeastern University Seattle

Introduction

With the rapid growth of online retail platforms like Amazon, customers often face difficulty navigating extensive product catalogs to find items relevant to their interests and purchase history. Effective recommendation systems can significantly improve user experience, reduce choice overload, and boost customer satisfaction. However, accurately predicting a customer's next purchase remains a challenging task.

In this research, we aim to address this challenge by developing and comparing multiple machine learning-based recommendation models to find the best fitting model that accurately suggests Amazon toy products to customers based on selected features. Leveraging neural network architectures and a pre-crawled Amazon toys dataset from Kaggle, our objective is to create personalized recommendations that effectively capture user preferences and historical buying patterns.

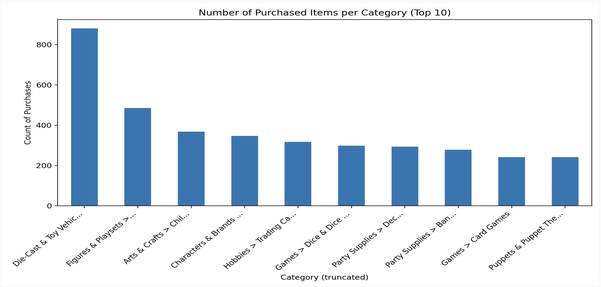
Goal and Motivation

Our aim was to build a robust recommender system for Amazon toy products using deep learning. With over 10,000 product listings and sparse user behavior data, traditional collaborative filtering is limited.
We explored **Linear Regression**, **content-based filtering** with **TF-IDF** and **deep learning** architectures to improve both precision and personalization.

Datasets

We used pre-crawled detailed dataset from **Kaggle** of toy products offered and purchased on **Amazon** to analyze purchases and create the recommendation model. The dataset includes multidimensional data the most important for our models being the product category, user ratings, price, and product name to support the predictive modeling for product recommendation.

We handled missing data by dropping certain fields where 90% of the field was missing, filling missing description and product_description observations with empty string, dropping rows where the category/sub-category is missing because that is the crucial piece of data needed and not able to back into. Missing price values were filled using median price per category.



Materials and Methods

1. Data Preprocessing

- **Text Normalization**: Product descriptions and metadata fields (e.g., product_name, description, product_information) were cleaned by removing HTML tags, punctuation, and stop words. Text was lowercased and lemmatized for consistency.
- **Structured Feature Handling**: Fields like price, average_review_rating, and number_of_reviews were normalized using Min-Max scaling. Products with missing critical values were excluded from training.
- **Deduplication**: As the dataset contained duplicate entries and noisy sellers' listings, we grouped records by uniq_id and retained the most complete entry.

2. TF-IDF + Cosine Similarity (Content-Based Filtering)

- We converted product descriptions into TF-IDF vectors to quantify the importance of words relative to the entire corpus.
- Pairwise cosine similarity was computed between a user's viewed/purchased product and all other items in the catalog.
- For each product interaction, we ranked and returned the top-k most similar items. This model was particularly effective when product descriptions were rich and unique.

3. Linear Regression Model

- We built a baseline linear regression model to predict the likelihood of co-purchase between two items using:
- Numerical features: price difference, rating difference, stock availability
- Categorical encodings: brand, sub-category
- The model was trained on product pairs, with a binary label indicating co-purchase (1) or not (0). While interpretable, it struggled with cold-start and sparse categories.

4. Deep Learning-Based Recommender (Prototype)

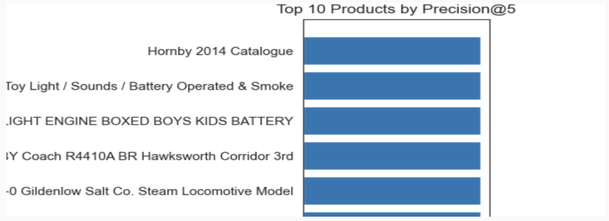
- **Embedding Layer**: For categorical variables like product category, brand, and manufacturer, we trained embeddings to capture latent semantics.
- **Dense Network**: Combined metadata features and learned embeddings were fed into fully connected layers with dropout and ReLU activations.
- **Loss Function**: Binary cross-entropy was used to predict the likelihood of co-purchase.
- The model was trained using mini-batch gradient descent with early stopping on validation loss.

5. Evaluation Metrics

- We used **MAP@5** (Mean Average Precision at 5) to evaluate recommendation quality based on ranked outputs.
- Additionally, we monitored **Precision@k**, **Recall@k**, and **F1-score** for different cutoff thresholds. Evaluation was performed on a 70-30 stratified split, ensuring that co-purchase relationships were preserved in both training and test sets.

Results

Our results show that **TF-IDF + Cosine Similarity** achieved a **Mean Average Precision@5 (MAP@5)** of **0.99**, indicating highly relevant top-5 recommendations. This performance demonstrates the strong discriminative power of content-based filtering when combined with descriptive product metadata.



Conclusions

Our study explored three primary approaches—TF-IDF with cosine similarity, linear regression, and deep learning—for building a product recommendation system tailored to the toy category on Amazon. Each method brings unique strengths and is suited for specific real-world scenarios:

1. **TF-IDF + Cosine Similarity: Fast, Lightweight, and Cold-Start Friendly**

- **Conclusion**: This method yielded the **highest precision (MAP@5 = 0.99)** by leveraging semantic information in product descriptions. Its strength lies in understanding product context without needing user behavior data.
- **Limitations**: Doesn't personalize recommendations. Similarity is based only on item content, not on user preferences or behavior.

1. **Linear Regression: Simple, Interpretable Baseline for Co-Purchase Prediction**

- **Conclusion**: While linear regression provided a reasonable baseline, it was limited in capturing non-linear relationships and contextual dependencies between products.
- **Limitations**: Poor performance on large or complex datasets with high-dimensional feature interactions. Also struggles with sparsity in co-purchase labels.

1. **Deep Learning-Based Recommender: High-Potential for Personalized Recommendations**

- **Conclusion**: While still in a prototype stage, the deep learning model demonstrated the ability to learn **latent relationships** between products using metadata and categorical features. It promises high scalability and personalization.
- **Limitations**: Requires large amounts of labeled data, hyperparameter tuning, and significant computational resources for training and inference.

Acknowledgement

- McAuley, J., Pandey, R., & Leskovec, J. (2015). **Inferring networks of substitutable and complementary products**. *Proceedings of the 21th ACM SIGKDD*.
- Aggarwal, C. C. (2016). *Recommender Systems: The Textbook*. Springer.
- Ricci, F., Rokach, L., & Shapira, B. (2015). *Recommender Systems Handbook*. Springer.
- Kaggle Dataset: Toy Products on Amazon Dataset

More Information

Ronne Kent – ronne.kent@northeastern.edu
Samridhi Singh – singh.samri@northeastern.edu