Fashion Recommendation System: A Hybrid Deep Learning Approach

Syracuse University

Sri Venkata Subhramanya Abhishek Namana, Himanshu Hegde, Punami Chowdary, Shaik Mohammed Nawazish Khalander

INTRODUCTION

Recommendation systems are essential for personalizing customer experience and driving business growth in e-commerce websites like Amazon and Shopify. Fashion e-commerce companies face unique challenges as they must take into account visual factors such as color and pattern, seasonality and unpredictable fashion trends. Our project addresses these challenges by developing a deep learning-based hybrid recommendation system. We developed a two-stage recommendation system for ranking and retrieval using a two-tower neural network architecture. To incorporate visual features, we generated image embeddings and added them as features to our item dataset, alongside other metadata features

DATA

Our data consists of 4 main files:
Images: A folder containing images for each item
Articles: Metadata for each item available for purchase
Customers: Metadata for each customer in the dataset
Transactions: Training data with purchase history, including dates and item quantities. Duplicate rows indicate multiple purchases of the same item.

FEATURE ENGINEERING

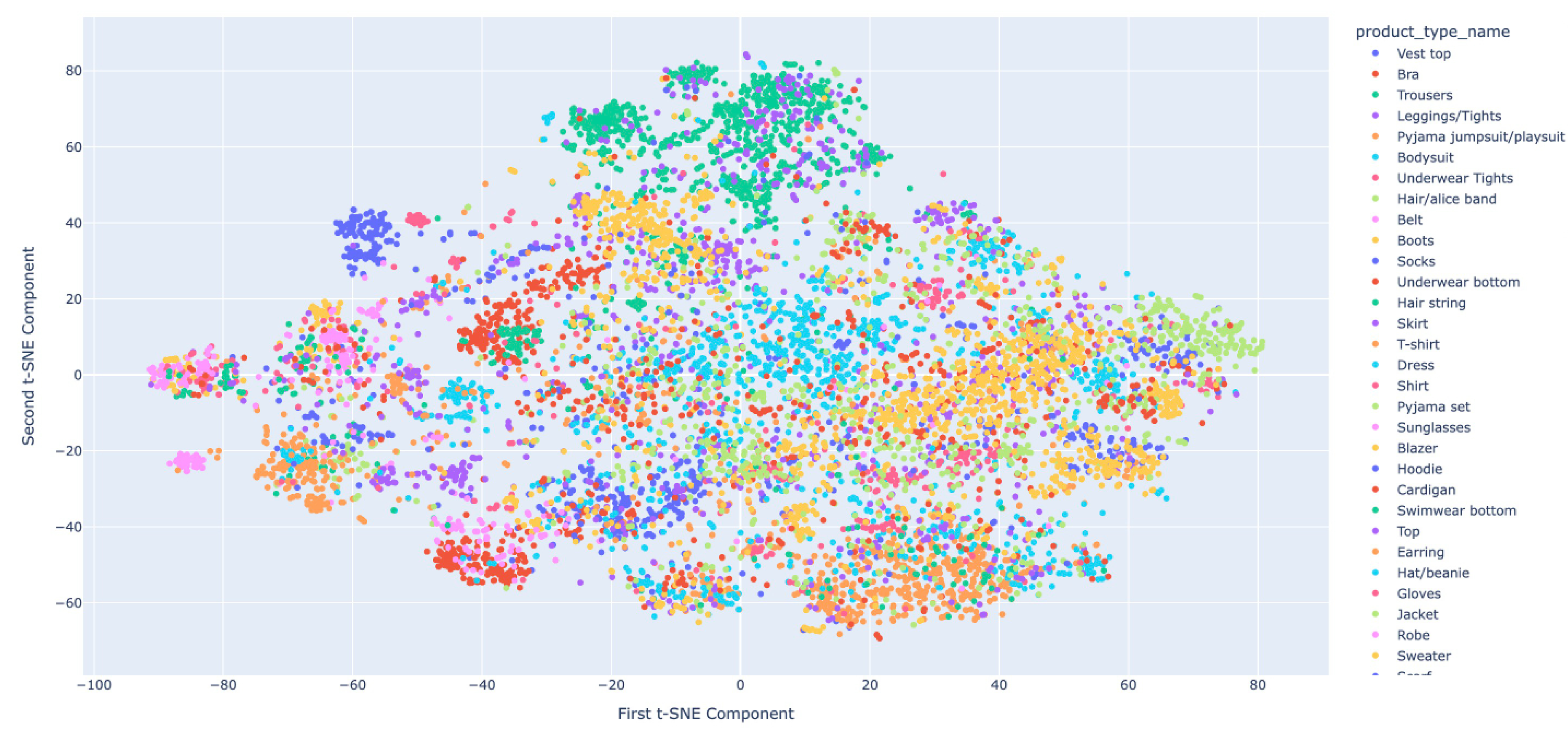
Feature Engineering was done to ensure maximum information extraction from the data into model features.

- Customer Features:
  - Purchase behavior metrics (count, frequency, spending patterns)
  - Recent activity indicators (last 30 days)
  - Customer metadata integration
  - Temporal engagement patterns
- Article (Item) Features:
  - Popularity metrics (sales count, unique customers)
  - Price statistics (mean, standard deviation, median)
  - Recent popularity indicators
  - Product metadata integration
  - Product (Image) Embedding Information
- Transaction Features:
  - Transaction-specific details
  - Negative Sampling: Generating negative labels for transaction.
- This is followed by:
  - Applying StandardScaler to continuous features
  - Implementing OneHotEncoder for categorical variables
  - Handles missing values based on data type
  - Converting numerical datatype to memory-efficient formats

METHODS

Image Embeddings:

To incorporate visual features such as color, pattern, and style, we used ResNet152 to create visual embeddings for the item images. PCA was used to reduce the dimensions of the resulting embeddings from 2048 to 256 while preserving 95% of variance. These embeddings were then added as features to our item dataset, alongside other item attributes.



t-SNE Visualization of ResNET152 embeddings of fashion items

Two Tower Recommender System:
Our two-tower recommender system uses dual neural networks - one tower processes customer features and another processes item features. Each tower independently learns to create embeddings for its respective inputs, and these embeddings are then compared (through dot product) to predict how well a user and item match. This architecture is particularly efficient for large-scale recommendations since embeddings can be pre-computed for all items and users separately.

We built the two-stage recommendation system using TensorFlow Recommenders (TFRS) to perform retrieval and ranking. The components are as follows:

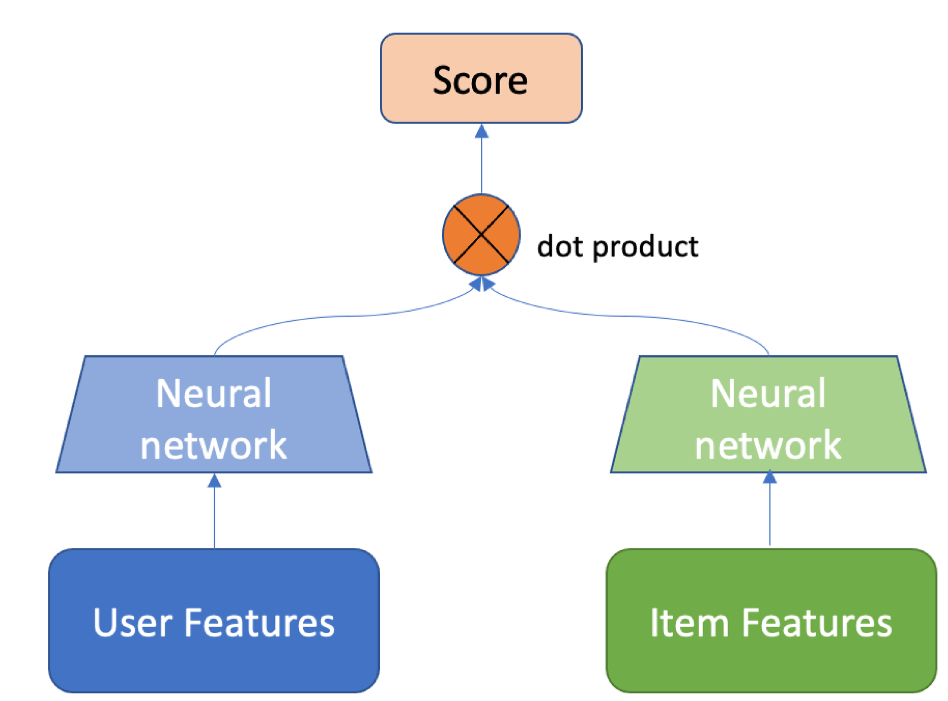
Feature Encoding: Customer and item features are encoded into two separate low-dimensional embeddings using Keras sequential layers with ReLU activation and normalization layers.
Ranking Model: A Customer Tower and an Item Tower create compact embeddings that capture the relationships between customer and item features. The combined embedding is passed through a neural network with dense layers to output a relevance score (0 to 1) using sigmoid activation.

Two-Stage Recommender:
Retrieval Step: Finds a broad set of relevant items for a customer using dot product similarity between embeddings.
Ranking Step: Narrows down retrieved items to the top recommendations using customer-item metadata

Loss Function:
- Retrieval Task - uses Contrastive Loss which teaches a model to tell if two things are similar or not. It does so by creating embeddings that are closer together for similar items.
- Ranking Task - uses Binary Cross Entropy. Relevance of an item for a user is framed as a binary classification problem (y=1 if relevant)
- Two Stage Recommender Model - combination of Retrieval Loss and Ranking Loss

Evaluation Metrics:
We used 3 evaluation metrics to evaluate the quality of our recommendations.
- Precision @ K - How many of the top K items we recommended were bought?
- Recall @ K - Of all the items the user bought, how many did we recommend?
- NDCG@ K (Normalized Discounted Cumulative Gain @K) - Scores the recommendations based on the order in which the items were recommended.

Stakeholder Metrics:
From a business perspective, fashion e-commerce websites can track engagement metrics such as Click Through Rate and Interactions per user, Revenue metrics such as Conversion rate and Revenue per user and Retention metrics such as User retention rate and Churn rate.



Two Tower Recommender System Architecture

RESULTS

Model Evaluation Metrics:
1) K = 10
Precision @ 10 = 18%
Recall @ 10 = 1%
NDCG @ 10 = 0.45

2) K = 20
Precision @ 20 = 15%
Recall @ 20 = 1%
NDCG @ 20 = 0.38

Interpretation
The Precision@10 of 18% indicates that nearly one in five recommended items are relevant, which drops slightly to 15% when extending to 20 recommendations. This drop is expected as it becomes harder to maintain precision with more recommendations. The NDCG scores (0.45 at k=10, dropping to 0.38 at k=20) suggest the system is placing relevant items in good positions, particularly in the top 10 results. The low recall (1% at both k=10 and k=20) could be due to several factors:
1)Fashion purchases are often spread across different categories, making it challenging to capture full purchase breadth.
2)Customers may have many purchases in the validation set, making it difficult to capture all their interests in just 10-20 recommendations.

Sample Recommendations for a regular HnM customer profile