

# NeuMF Implementation Plan for Book Recommendation System

## Phase 1: Data Preparation

### 1. Format User-Item Interactions

- Transform your ETL-processed data into user-book interaction format
- Map UserIDs and BookIDs to continuous integer indices
- Store these mappings for later inference
- Analyze rating distribution to determine appropriate threshold for binary conversion
- Consider the sparsity in your dataset (many zero ratings) when making format decisions

### 2. Convert to Binary Feedback

- Define positive interactions (e.g., ratings above 7 on your 0-10 scale)
- Use your `explicit_rating` column as a starting point
- Experiment with different thresholds if necessary
- Calculate interaction density to understand your data's sparsity level
- Document distribution patterns for later interpretation

### 3. Create Data Splits

- Implement time-based train/validation/test splits (70/10/20)
- Ensure all users appear in all splits to avoid cold-start scenarios
- Structure data for efficient loading during training
- Verify split statistics match expectations

### 4. Negative Sampling Strategy

- Design sampling approach for unobserved (user, book) pairs
- Consider weighted sampling based on book popularity
- Create separate sampling methods for training and evaluation
- Aim for 4-8 negative samples per positive during training
- Use 100 negative samples per positive for evaluation

## Phase 2: Feature Engineering Decisions

### 1. Core Interaction Features

- Include the basic user and item IDs as primary inputs
- Convert book ratings to binary indicators (interacted/not interacted)
- Consider rating values as weights rather than binary if appropriate

### 2. Leveraging ETL-Generated Features

- Evaluate incorporating your engineered features:

- `Book_per_rating_encoded`: Book's average rating
- `Author_per_rating_encoded`: Author's average rating
- `Publisher_per_rating`: Publisher's average rating
- Consider these as auxiliary inputs to supplement the main model
- Test model versions with and without these features
- These can be especially valuable for addressing cold-start issues

### 3. Additional Feature Considerations

- User age can provide demographic context
- Book publication year for temporal patterns
- Consider user activity level and book popularity as additional signals
- Balance feature richness with model complexity

## Phase 3: NeuMF Model Implementation

### 1. Architecture Design

- Single model with two internal pathways
- User and item embedding layers for each pathway
- Element-wise product path (for learning linear interactions)
- Multi-layer perceptron path (for learning non-linear interactions)
- Fusion layer to combine both pathways
- Output layer with sigmoid activation for binary prediction

### 2. Model Configuration

- Define embedding dimensions for both pathways
- Set MLP layer sizes with appropriate activation functions
- Add regularization to prevent overfitting
- Configure loss function (binary cross-entropy)
- Set up optimizer with appropriate learning rate

### 3. Training Strategy

- Define batch generation process
- Implement negative sampling during training
- Configure early stopping based on validation metrics
- Set up learning rate scheduling
- Create model checkpointing to save best versions
- Monitor training progress with appropriate metrics

## Phase 4: Evaluation Framework

### 1. Metrics Implementation

- Implement ranking metrics:
  - Hit Ratio @ k (HR@k)

- Normalized Discounted Cumulative Gain @ k (NDCG@k)
- Area Under ROC Curve (AUC)
- Use standardized evaluation procedure for consistency

## **2. Performance Assessment**

- Compare against baseline methods
- Analyze performance across different user segments
- Evaluate how the model handles cold-start situations
- Analyze impact of feature engineering choices

## **3. Hyperparameter Optimization**

- Systematically test key parameters:
  - Embedding dimensions
  - MLP layer configurations
  - Learning rate and batch size
  - Negative sampling ratio
- Document performance implications of different choices

# **Phase 5: Recommendation Generation**

## **1. Inference Pipeline Design**

- Build function to generate top-N recommendations for users
- Implement batch prediction for efficiency
- Convert internal indices back to original IDs
- Filter out previously interacted items from recommendations

## **2. Practical Application**

- Create simple interface for testing recommendations
- Implement evaluation of recommendation quality
- Document model limitations and strengths
- Suggest potential improvements for future work

This plan provides a comprehensive yet focused approach for implementing a Neural Collaborative Filtering model for your book recommendation system. It leverages your ETL-processed data while addressing the unique challenges of your book dataset, particularly the rating sparsity issues.