推荐系统算法实现与优化方案

基于您提供的数据集分析，我推荐按照从基础到高级的顺序实现以下推荐算法，这样可以逐步提升性能并解决数据特有的挑战。

1. 基础基线算法

首先实现简单的非个性化方法作为基准：

- **全局平均评分**：所有预测使用训练集整体平均评分
- **用户均值/物品均值**：分别使用用户平均评分和物品平均评分进行预测
- **基于偏置的方法**：结合全局均值、用户偏置和物品偏置的预测

2. 基于邻域的协同过滤

- **基于物品的协同过滤（ItemCF）**：对于高稀疏数据更有效，可尝试不同相似度计算方法：
  - 余弦相似度
  - 皮尔逊相关系数
  - 调整余弦相似度
- **基于用户的协同过滤（UserCF）**：作为对比，通常在这类数据集上表现不如 ItemCF

3. 矩阵分解方法

这类方法特别适合您的高稀疏数据集：

- **基本 SVD（奇异值分解）**：降维并捕获隐藏特征
- **偏置 SVD（Biased SVD）**：引入用户/物品偏置，处理评分偏向正面的特点
- **非负矩阵分解（NMF）**：限制因子为正，提高可解释性
- **SVD++**：结合隐式反馈，提高预测准确度

4. 高级模型

- **因子分解机（FM）**：能更好地建模特征交互
- **神经网络协同过滤（NCF）**：结合深度学习和传统推荐方法
- **LightGCN**：基于图神经网络的推荐方法，适合处理稀疏数据

5. 处理冷启动问题的策略

针对测试集用户数大于训练集用户数的情况：

- **基于内容的混合方法**：结合物品属性（如有额外特征）
- **基于热门物品的推荐**：对新用户推荐热门物品
- **元学习方法**：快速适应新用户的偏好

6. 集成与混合策略

- **加权混合模型**：结合多种算法的优点
- **堆叠模型**：使用一个模型的输出作为另一个模型的输入
- **多目标优化**：同时考虑准确性和多样性

实施路线建议

1. **先实现基线方法** 确立性能基准
2. **重点实现 ItemCF 和 BiasedSVD**，这两种算法很可能在您的数据集上表现良好
3. **逐步尝试更复杂的模型**，每次优化时分析模型的问题并有针对性地改进
4. **单独处理冷启动问题**，为测试集中的新用户设计专门的推荐策略

您可以使用均方根误差(RMSE)、平均绝对误差(MAE)等指标评估预测准确性，并使用覆盖率、多样性等指标衡量推荐质量的其他方面。