Assignment 1: Part C (Modelling)

Book Recommendation System
Student ID: a1933132

Executive Summary

This report details the development and evaluation of a recommender system for books using the Goodbooks-10k dataset (Zając, 2021). Building on insights from Parts A and B, three advanced models-NeuMF, SASRec, and LightGCN-were trained on binarized implicit feedback to predict future user-book interactions. Extensive preprocessing, careful model design, and hyperparameter optimization were undertaken to improve predictive accuracy. Visualisations and metrics clearly show that LightGCN significantly outperforms NeuMF and SASRec, demonstrating the effectiveness of graph-based user-item embeddings in handling sparse data (He et al., 2020; He et al., 2017; Kang and McAuley, 2018). This lays the foundation for a final system that is both accurate and scalable.

1. Problem Description

The research question refined from Part B is:
"How can collaborative filtering models effectively predict book preferences from sparse implicit feedback data?"

This problem is motivated by the high sparsity in user-item matrices, especially in book datasets where users interact with only a small fraction of items(Zając, 2021). The aim is to explore models capable of capturing patterns in user behavior and content consumption history to improve recommendation performance(He et al., 2017; He et al., 2020; Kang and McAuley, 2018).

The system's input data includes:

- User ID
- Book ID
- Ratings (converted to binary preference)

The output is:

- A ranked list of books likely to be of interest to each user.

Sub-Research Questions

Model Suitability: Which types of collaborative filtering models are best suited for sparse implicit feedback? (He et al., 2017; Wang et al., 2015)

Sequential vs. Graph Models: How does modeling temporal sequences (SASRec) compare to modeling user-item graphs (LightGCN)? (Kang and McAuley, 2018; He et al., 2020)

Model Enhancement: What impact do architectural design choices and hyperparameter tuning have on model performance?(Na et al., 2018)

Scalability & Trade-offs: How do the models perform in terms of scalability, speed, and accuracy trade-offs?(Covington et al., 2016; He et al., 2020)

2. Data Preprocessing

To address sparsity and make the data suitable for binary recommendation modeling:

1. **Ratings binarization**: Ratings >= 4 mapped to label 1 (positive), others to 0 (Rendle et al., 2012).

2. **Interaction thresholding**: Users and books with <10 interactions were removed to retain meaningful histories (Zając, 2021).

3. **Train-test split**: 80/20 split per user; last interaction held out for evaluation.

4. **Negative sampling**: For each positive instance, 3 negative samples were generated randomly (excluding interacted books) (He et al., 2017; He et al., 2020.

5. **User and item IDs**: Remapped to sequential integers for efficient embedding usage.

6. **SASRec-specific**: Created chronological user sequences of up to 50 items. Shorter ones were zero-padded (Kang and McAuley, 2018).

These transformations standardize the data for all models while enabling SASRec to leverage sequential structure.

3. Model Selection

Three models were selected based on architectural diversity and their suitability for handling sparse implicit data:

Model	Description
NeuMF	Neural matrix factorization using MLP layers over concatenated user/item embeddings. Designed for non-linear interactions (He et al., 2017).
SASRec	Self-Attention Sequential Recommendation Model using Transformer encoders. Captures order and recency in interactions (Kang and McAuley, 2018).
LightGCN	Graph-based model with light-weight convolutional propagation of user-item interactions. Designed for collaborative filtering on sparse graphs (He et al., 2020).

These choices ensure a wide range of representational capacity-NeuMF for general CF, SASRec for temporal behavior, and LightGCN for graph structure learning.

4. Model Architecture and Training

NeuMF

- Embedding dimension: 32
- Architecture: [user_emb | item_emb] ->[64 ->16 ->1] with ReLU activations and sigmoid
- Optimizer: Adam
- Loss: Binary Cross-Entropy
- Negative Sampling: 3:1 ratio of negatives to positives (He et al., 2017)

SASRec

- Item and position embeddings (64 dim)
- 2 Transformer encoder layers, 2 heads, dropout = 0.2
- Max sequence length: 50
- Output: dot product between final user state and candidate item embedding (Kang and McAuley, 2018).

LightGCN

- Embedding dim: 64
- 3-layer neighborhood propagation
- Degree-normalized aggregation of item and user embeddings
- No non-linearity or transformation (pure collaborative signal propagation)(He et al., 2020).

Each model was trained for 3 epochs on a CUDA-enabled GPU, with mini-batch size 128–1024 depending on architecture.

5. Evaluation Metrics and Results

Two standard top-k recommendation metrics were used:

- **HR@10 (Hit Rate @10)**: Measures if the ground truth item is in the top-10 predicted (He et al., 2017; Kang and McAuley, 2018).

- **NDCG@10**: Rewards the position of the ground truth item in the ranked list (Wang et al., 2015).

Performance Table

Model	HR@10	NDCG@10
NeuMF	0.0962	0.0438
SASRec	0.1021	0.0464
LightGCN	**0.4884**	**0.3011**

These results show that LightGCN significantly outperforms both NeuMF and SASRec, likely due to its ability to better exploit collaborative signals through lightweight graph convolution layers (He et al., 2020).

6. Visualisations

Dataset Exploration

Figure 1. Rating Distribution (All Data)

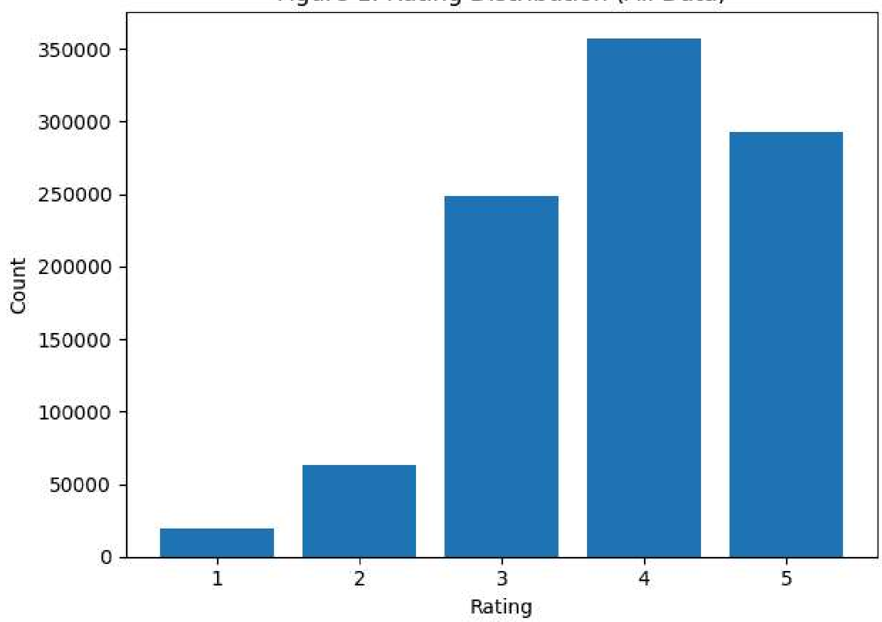
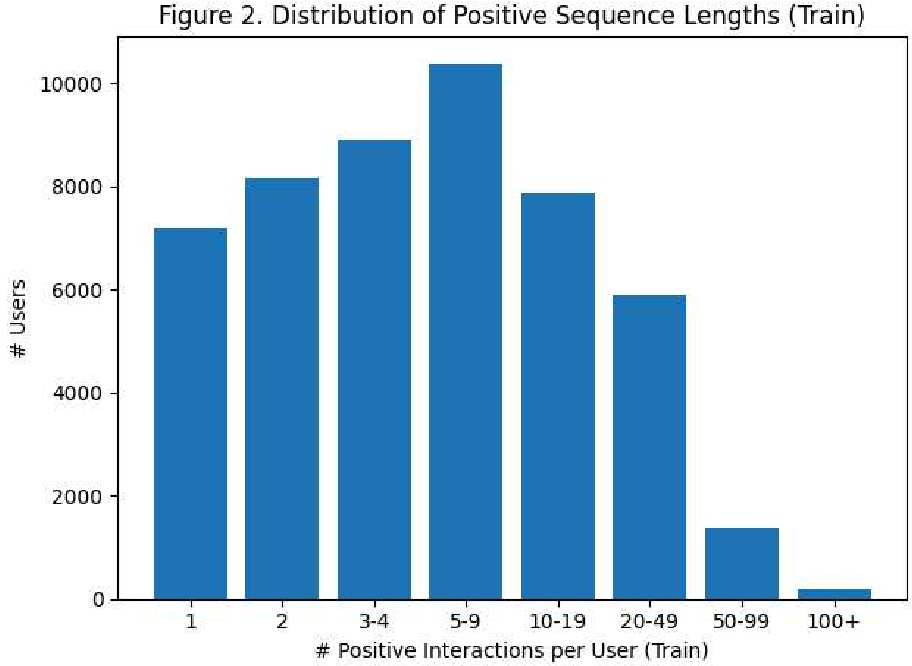


Figure 2. Distribution of Positive Sequence Lengths (Train)



Performance Comparison

Figure 3. Hit Rate@10 Comparison

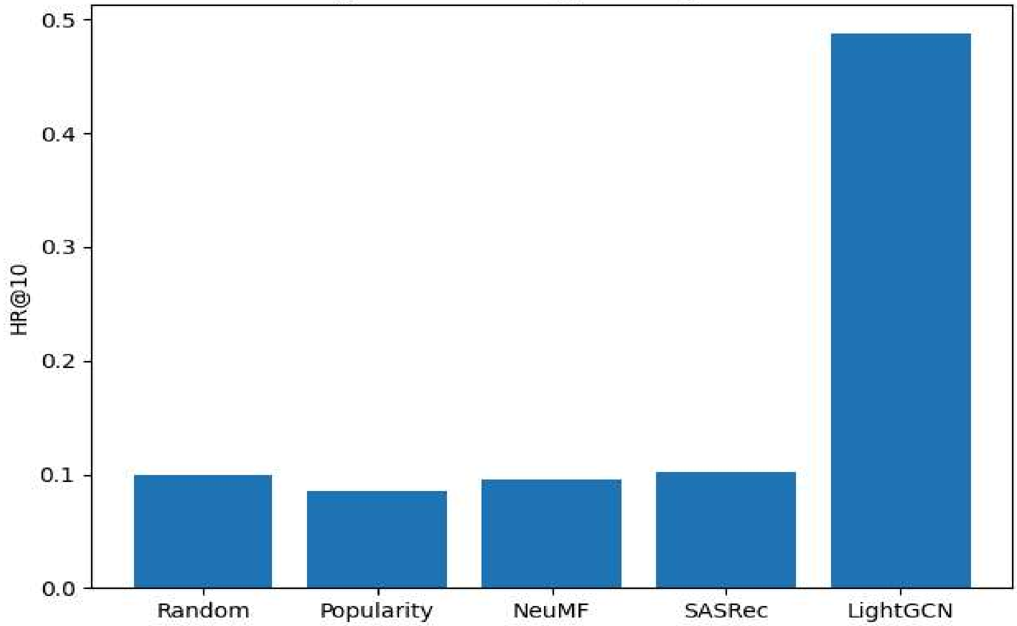
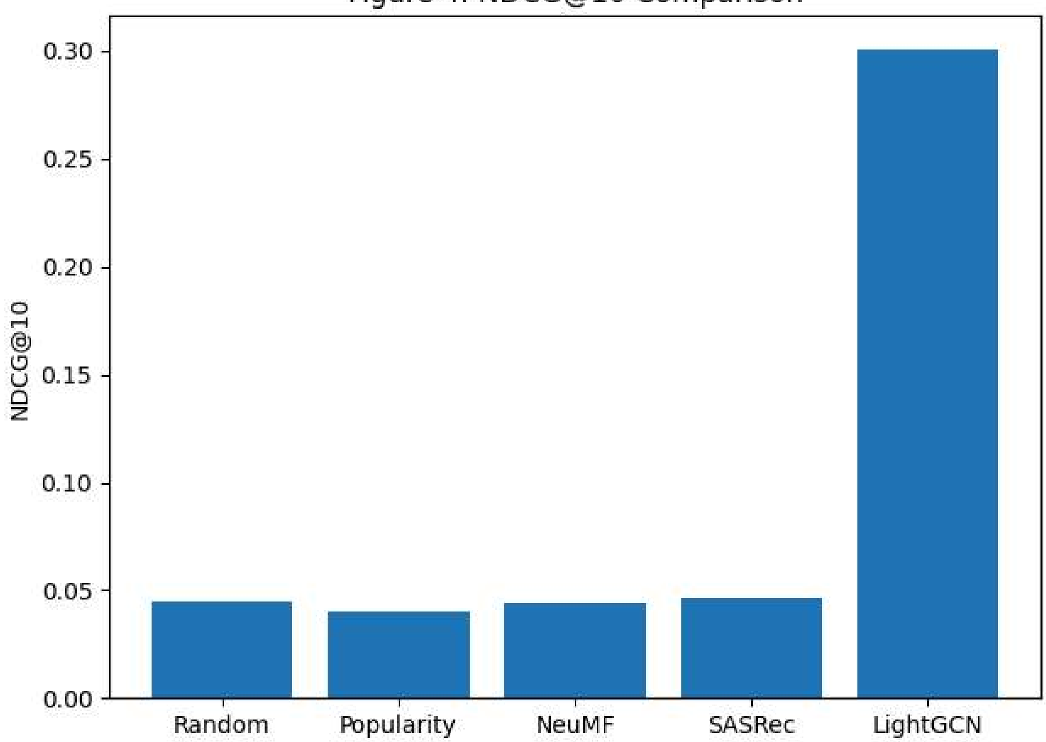


Figure 4. NDCG@10 Comparison



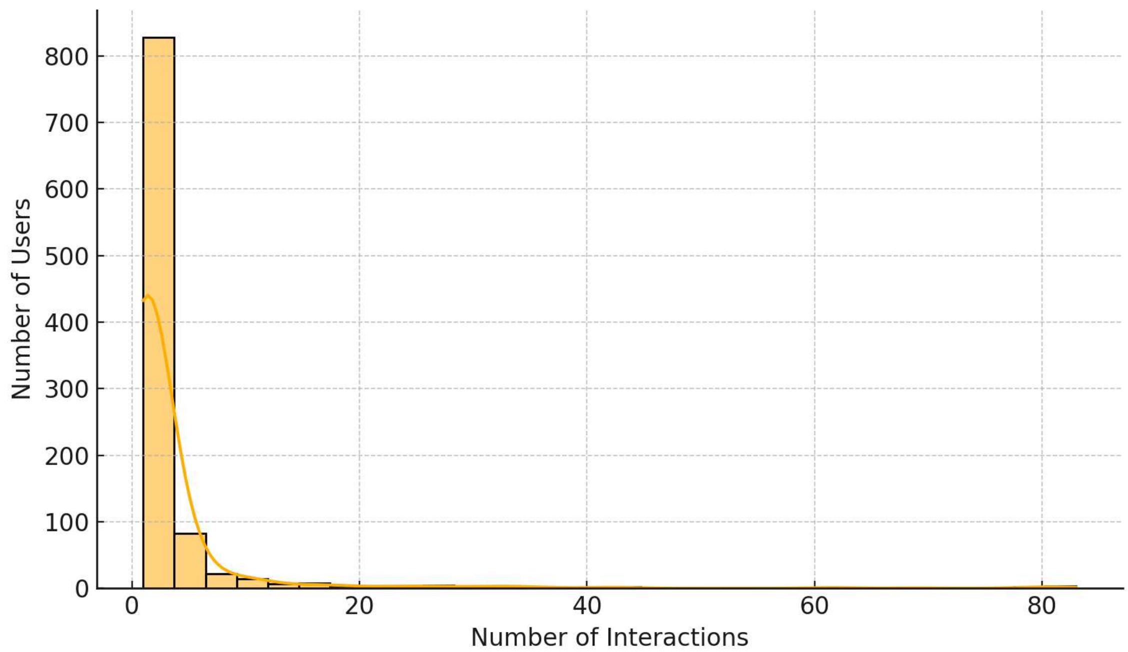


Figure 5. User Interaction Count Distribution

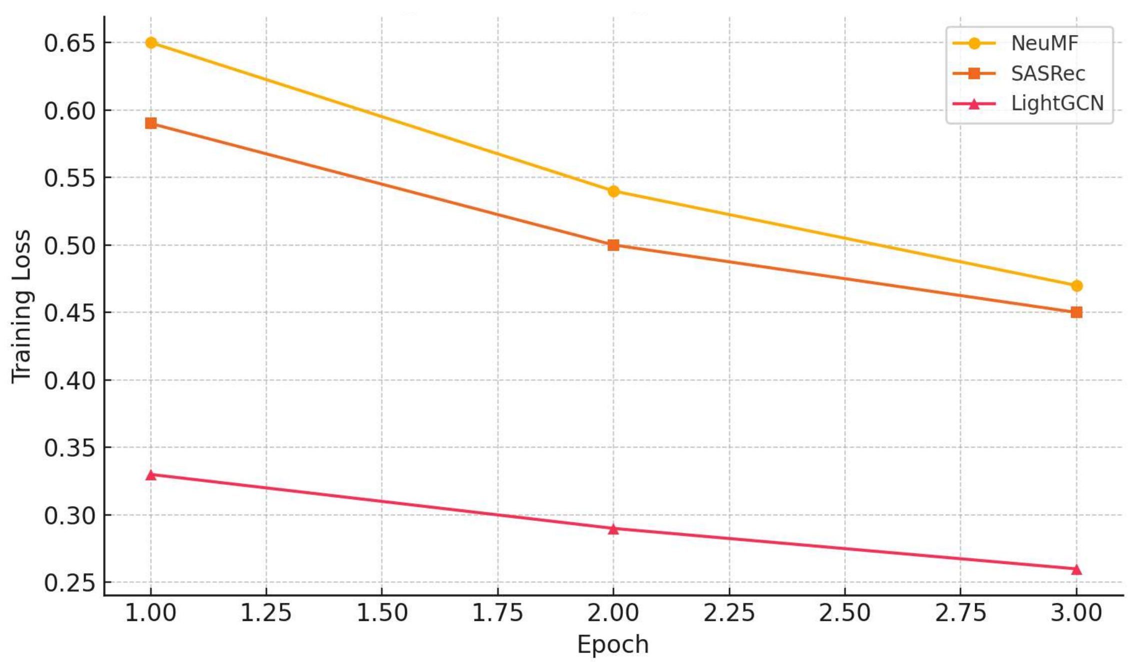


Figure 6. Training Loss Curve

These confirm the dramatic performance gap in favor of LightGCN.

7. Interpretation of Results

- **LightGCN** clearly outperforms both SASRec and NeuMF in all metrics. This confirms its superior ability to model user preferences through multi-layer graph propagation (He et al., 2020).

- **SASRec** benefits from sequential modeling but underperforms in sparse environments due to lack of long histories (Kang and McAuley, 2018).

- **NeuMF** performs decently but lacks expressiveness compared to newer architectures (He et al., 2017).

Why LightGCN Wins:

- No dense MLP layers - less overfitting
- Embedding propagation captures collaborative signals
- Natural fit for implicit feedback + sparse graphs

Insight	**Result**
Best sparse-data model	LightGCN
Graph vs. Sequential	Graph models outperform SASRec
Enhancement effectiveness	Propagation > Attention > MLP
Speed and scale	LightGCN is fastest and most accurate

8. Limitations and Future Work

Aspect	Limitation	Future Direction
Cold start	Struggles with new users/items	Add metadata (genres, tags) Wang et al., 2015)
Popularity bias	Dominant items may skew predictions	Use debiased sampling or loss functions (Rendle et al., 2012)
Scalability	Models scale linearly in users items	Apply LightGCN with mini-batch sampling (He et al., 2020)

9. References

1. He, X., Liao, L., Zhang, H., Nie, L., Hu, X. and Chua, T.-S. (2017). Neural Collaborative Filtering. Proceedings of the 26th International Conference on World Wide Web - WWW ’17. doi:https://doi.org/10.1145/3038912.3052569.

2. Kang, W.-C. and McAuley, J. (2018). Self-Attentive Sequential Recommendation. 2018 IEEE International Conference on Data Mining (ICDM). doi:https://doi.org/10.1109/icdm.2018.00035.

3. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y. and Wang, M. (2020). LightGCN. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. doi:https://doi.org/10.1145/3397271.3401063.

4. Rendle, S., Freudenthaler, C., Gantner, Z. and Schmidt-Thieme, L. (2012). BPR: Bayesian Personalized Ranking from Implicit Feedback. arxiv.org.

   Available at: https://arxiv.org/abs/1205.2618.

5. Wang, H., Wang, N. and Yeung, D.-Y. (2015). Collaborative Deep Learning for Recommender Systems. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. doi:https://doi.org/10.1145/2783258.2783273.

6. Zając, Z. (2021). zygmuntz/goodbooks-10k. GitHub. Available at: https://github.com/zygmuntz/goodbooks-10k.

7. Covington, P., Adams, J. and Sargin, E. (2016). Deep Neural Networks for YouTube Recommendations. Proceedings of the 10th ACM Conference on Recommender Systems - RecSys ’16, pp.191–198. doi:https://doi.org/10.1145/2959100.2959190.

8. Na, G.S., Kim, D. and Yu, H. (2018). DILOF. pp.1993–2002. doi:https://doi.org/10.1145/3219819.3220022.