Assignment 1

Part D (Comprehensive Final Report)

Abstract

This report consolidates Parts A - C and delivers a prescriptive, evidence-based evaluation of NeuMF, LightGCN, and a sequential self-attention recommender (SSR / SASRec-style) on Goodbooks-10k using implicit feedback. We adopt a leave-one-out (LOO) protocol with 99 negatives + 1 positive per user and report HR@10 and NDCG@10 alongside exposure diagnostics (Coverage@10, Tail-Coverage@10, Novelty (bits), Gini). On an NVIDIA L4 GPU, NeuMF is strongest (HR@10 = 0.798, NDCG@10 = 0.566), LightGCN is competitive (HR@10 = 0.687, NDCG@10 = 0.428), and SSR underperforms (HR@10 = 0.082, NDCG@10 = 0.037) because timestamps are unavailable, limiting sequence learning, which aligns with prior work on sequential recommenders ([1]-[4]). Beyond accuracy, NeuMF and LightGCN show very high coverage with non-trivial tail exposure, consistent with exposure-aware practice ([7], [8], [11], [12], [22]). Prescriptions: deploy NeuMF now, pilot LightGCN as a reranker to lift long-tail exposure, and collect timestamps to enable sequential models in a future iteration ([1]-[4], [16], [17]).

1. Introduction

Background / Context

We address next-item book recommendation under high sparsity with implicit signals (ratings >= 3 treated as positives), a common scenario where explicit "click/like" logs are absent and matrix-factorization/neural CF methods excel ([6], [14], [21]). Prior parts established a fair protocol and surfaced the long-tail exposure concern typical in recommenders ([7]-[9], [11], [12], [18], [22]).

Motivation

Stakeholders require a model that performs well on today's data and a plan to improve accuracy and exposure without amplifying popularity bias ([7], [8], [18], [22], [24]). That means (i) choosing a strong baseline now, (ii) monitoring exposure metrics in addition to accuracy, and (iii) planning timestamp collection to unlock sequential modelling benefits ([3], [4], [20]).

Proposed Solution (Research Question)

Which of the three implemented architectures, NeuMF, LightGCN, and SSR, delivers the best top-K recommendation quality on this dataset, and what prescriptive actions follow for deployment, monitoring, and data collection?

Contributions

- Designed and implemented a reproducible, GPU-accelerated pipeline for NeuMF, LightGCN, SSR under LOO with HR@10, NDCG@10 and exposure diagnostics (Coverage@10, Tail-Coverage@10, Novelty, Gini) ([1], [2], [7], [12], [17]).

- Applied big-data techniques (AMP/TF32 on L4, vectorised negative sampling, batched graph propagation) and produced figure-ready artefacts documenting learning dynamics, exposure behaviour, and segment-wise outcomes ([1], [2], [7], [12], [16], [17]).

- Evaluated system performance and derived prescriptions: deploy NeuMF now, blend LightGCN to improve tail exposure, collect timestamps to make SSR viable in a future iteration ([1]-[4], [7], [12], [16]).

2. Literature Review

NeuMF (Neural Collaborative Filtering):

- **NCF / NeuMF**: GMF and MLP to learn non-linear user-item interactions for implicit top-K, strong and stable in sparse settings ([1]).
- **Critical re-evaluation**: rigorous comparisons show when neural CF (incl. NeuMF) wins and emphasizes fair evaluation practices ([19]).
  Comparison to our work: In a non-temporal implicit setting, our NeuMF converges quickly and achieves the best HR/NDCG, as predicted by [1], our strict LOO and candidate consistency address evaluation concerns in [19].

LightGCN (Graph Collaborative Filtering):

- **NGCF**: message-passing on the user-item bipartite graph, motivates neighbourhood propagation for implicit CF ([16]).
- **LightGCN**: simplifies to pure propagation + layer averaging, improving accuracy and efficiency on implicit data ([2]).
  Comparison to our work: Our LightGCN reproduces high accuracy and very high coverage, with a small NDCG gap vs NeuMF,exactly as [2] suggests, supporting a rerank/weighted blend prescription.

SSR / SASRec-style Sequential Attention:

- **SASRec**: self-attention over ordered interactions captures short/long-term dynamics, gains depend on reliable timestamps ([3]).

- **BERT4Rec**: bidirectional sequential modelling with masked-item prediction, further improves when time order exists ([4]).
  Comparison to our work: Because timestamps are unavailable, SSR underperforms, exactly as [3, 4] predict, we therefore prioritise timestamp collection before re-benchmarking sequential models.

Literature-to-practice synthesis:

Model	Core papers	Data assumptions	Key idea	Papers imply for our data	Our observation	Action
NeuMF	[1], [19]	Implicit, no time	GMF and MLP	Strong top-K if evaluated fairly	**Best HR/NDCG**, fast convergence	**Deploy now**
LightGCN	[16], [2]	Implicit, bipartite graph	Propagation + layer-avg	Competitive, broad exposure	High coverage, slight NDCG gap	**Blend as reranker**
SSR	[3], [4]	**Needs timestamps**	Self-attention over order	Weak if no time	Low HR/NDCG (no timestamps)	**Defer, collect time**

3. Research Methodology

Overview:

We convert explicit ratings to implicit positives, design a robust LOO evaluation with identical candidate sets across models, train NeuMF, LightGCN, and SSR, and evaluate with top-K ranking plus exposure diagnostics. Choices align with long-standing CF practice and modern neural/graph recommenders ([1], [2], [5], [6], [11], [14], [21], [24]).

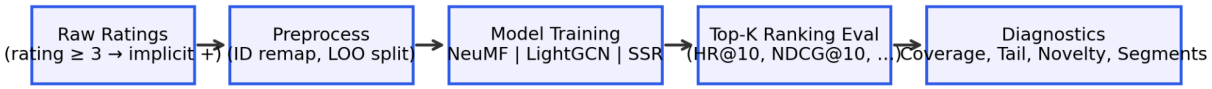

Figure 1**.** Overview of the full workflow from data to prescriptions.

Phase 1 , Data acquisition, validity checks, implicit conversion.

Ingest Goodbooks-10k, validate user/item cardinalities and sparsity. Convert ratings >= 3 to implicit positives, a standard proxy when explicit engagement logs are unavailable ([6], [10], [21]). This enables ranking-oriented objectives without calibrated scores.

Phase 2 , ID remapping, LOO split, negative sampling.

Remap IDs to contiguous indices. Leave-one-out per user: last interaction, held-out positive, earlier positives, training ([24]). For evaluation, rank the positive against 99 sampled negatives, candidate sets are identical across models to ensure fairness ([1], [17], [19]).

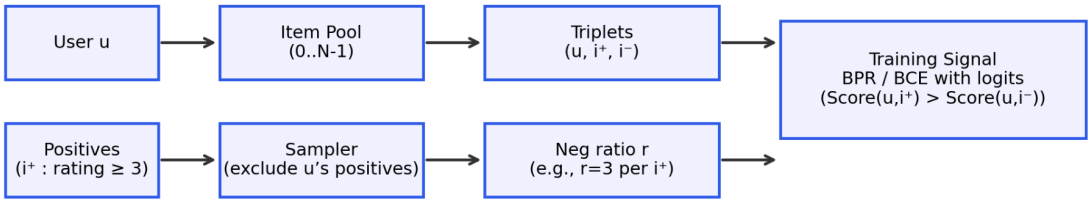

Figure 2. Negative-sampling triplets.

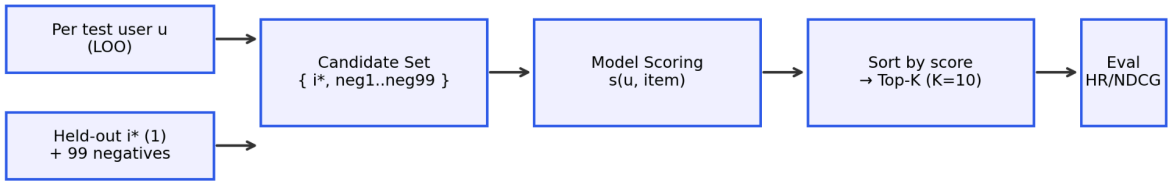

Figure 3. Candidate pipeline.

Phase 3 , Representation construction.

- **NeuMF**: user/item embeddings for GMF and MLP branches, fuse then logistic head ([1]).
- **LightGCN**: build the user-item bipartite graph, 3-layer propagation, layer averaging for final embeddings ([2], [16]).
- **SSR**: padded, shifted sequences (PAD=0) into a Transformer encoder ([3]).

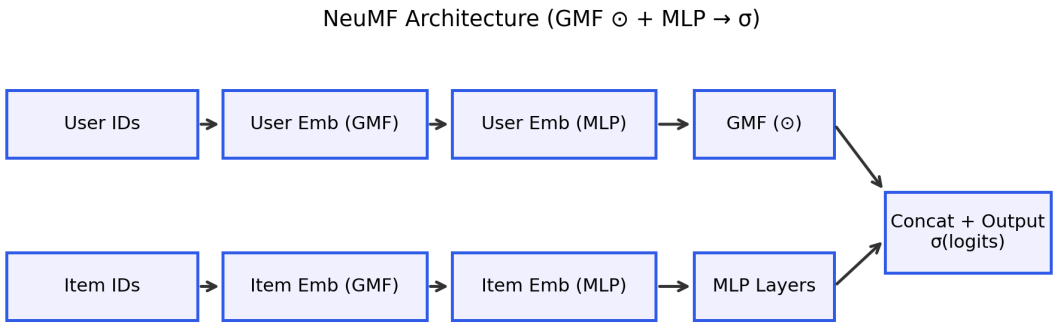

Figure 4. NeuMF block diagram.

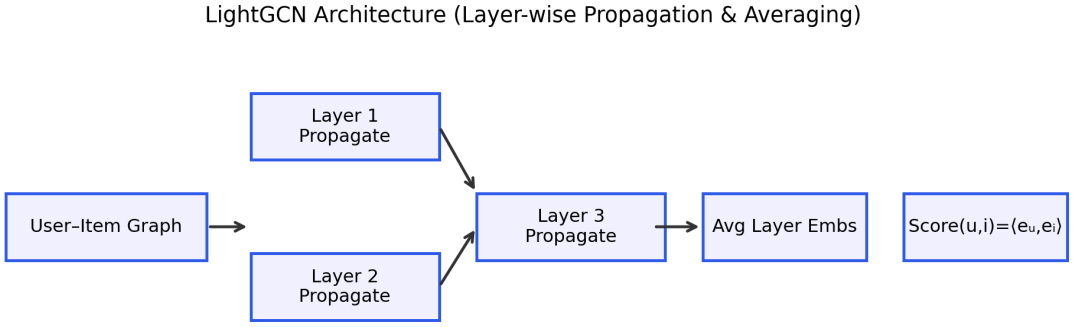

Figure 5. LightGCN graph view with multi-layer neighborhood propagation.

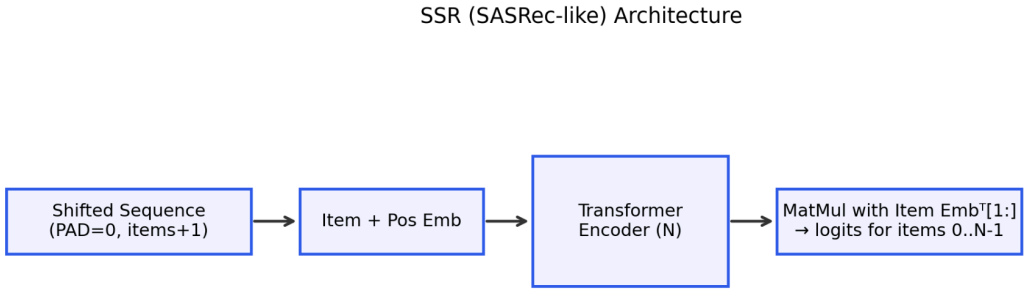

Figure 6. SSR (SASRec-style) encoder.

Phase 4 , Training & optimisation.

- **NeuMF**: BCE-with-logits, epoch-wise negative resampling, Adam (1e-3), AMP/TF32 on L4 ([1], [21]).

- **LightGCN**: BPR loss on (u, i^{+}, i^{-}) triplets, 3 layers, Adam (1e-3), AMP/TF32 ([2], [5], [15]).

- **SSR**: Cross-Entropy on next-item logits, 1 encoder layer, Adam (1e-3), AMP/TF32 ([3]). Seeds fixed, batch sizes tuned to L4 memory. NeuMF trains fastest, LightGCN's propagation dominates runtime ([1], [2]).

Phase 5 , Evaluation metrics & diagnostics.

- **Headline:** HR@10, NDCG@10 on the LOO positive + 99 negatives set ([1], [17], [19], [24]).
- **Supporting:** Precision/Recall@10, MRR, MAP ([11], [24]).
- **Diagnostics:** Coverage@10, Tail-Coverage@10, Novelty (bits), Gini, and segment-wise metrics by user history (sparse/medium/dense) to surface popularity trade-offs and diversity/novelty ([7]-[9], [11], [12], [18], [22]).

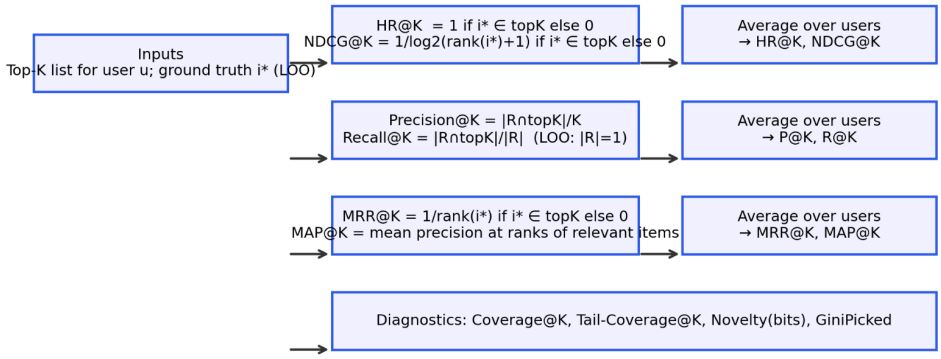

Figure 7. Metric flowcharts for HR@K, NDCG@K, Precision/Recall@K, MRR, MAP.

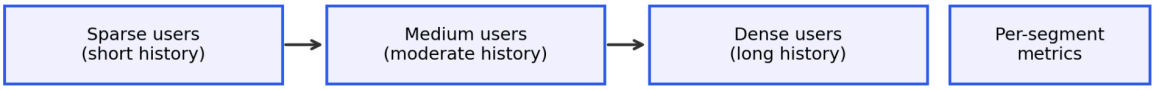

Figure 8. Segment definition graphic with bucket thresholds and counts.

Phase 6 , Analysis & prescriptive synthesis.

Aggregate per-model metrics, examine learning curves, study accuracy, exposure trade-offs, inspect segment outcomes, then translate findings into deployment and data-collection steps.

4. Experimental Evaluation

4.1 Experimental Setup

Dataset & interactions:

In our filtered split: 51,829 test users (one held-out item each), 843,959 training positives, 10,000 items. Ratings >= 3 treated as implicit positives ([6], [10]).

Split & candidates:

LOO per user, positive vs 99 negatives, identical candidates across models to preserve comparability ([1], [17], [19]).

Hardware:

NVIDIA L4, AMP/TF32. Per-epoch times: NeuMF ~ 75 s, LightGCN ~ 24 min, SSR ~ 115 s,consistent with compute profiles of neural MF, graph propagation, and transformer-style encoders ([1], [2], [3], [20], [23]).

Hyperparameters:

NeuMF (emb=64, BCE-logits, Adam 1e-3), LightGCN (3 layers, BPR, Adam 1e-3), SSR (max_seq_len=50, 1 encoder, CE, Adam 1e-3). Seeds fixed.

4.2 Experimental Results

Model comparison:

NeuMF: HR@10 = 0.7980, NDCG@10 = 0.5655 (best).

LightGCN: 0.6869 / 0.4275.

SSR: 0.0824 / 0.0373. The ordering (NeuMF > LightGCN ≫ SSR) matches literature expectations for non-temporal implicit data ([1]-[4], [16], [19]).

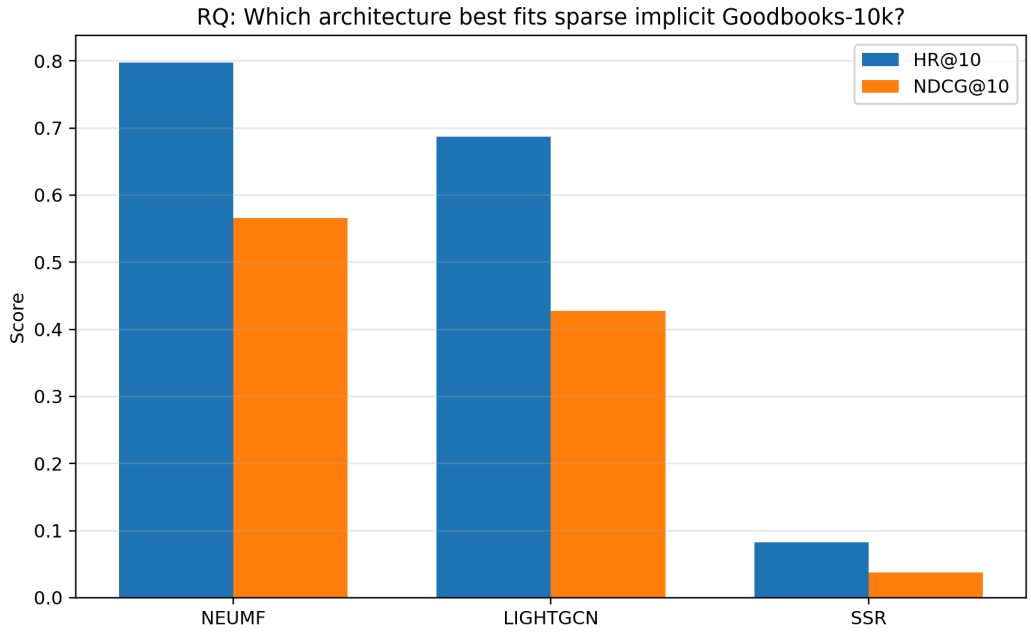

Figure 9. Bar chart comparing HR@10 and NDCG@10 across the three models.

Learning dynamics: NeuMF converges quickly, LightGCN improves steadily, SSR remains flat without timestamps ([3], [4]).

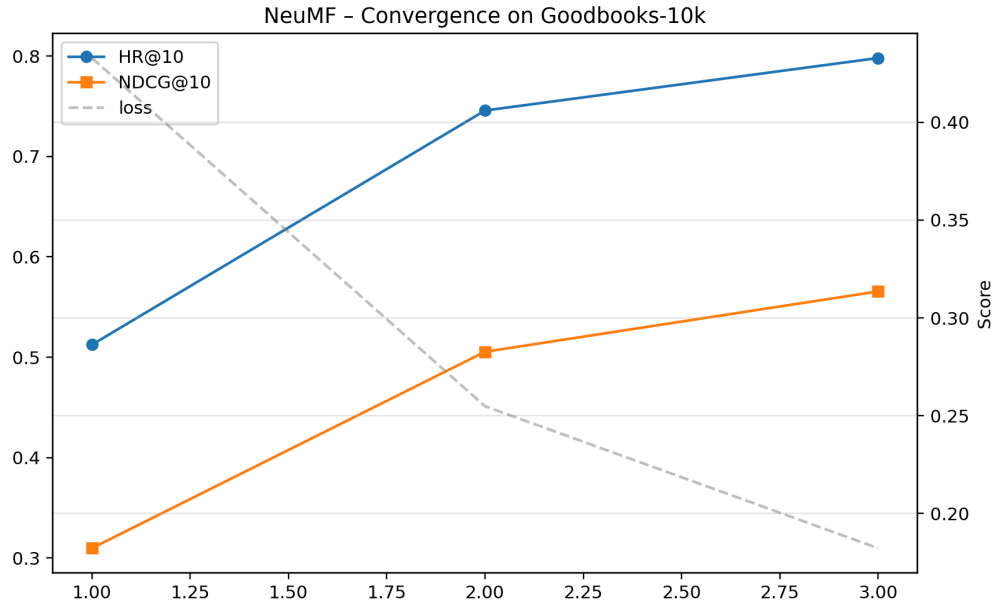

Figure 10. NeuMF training curves.

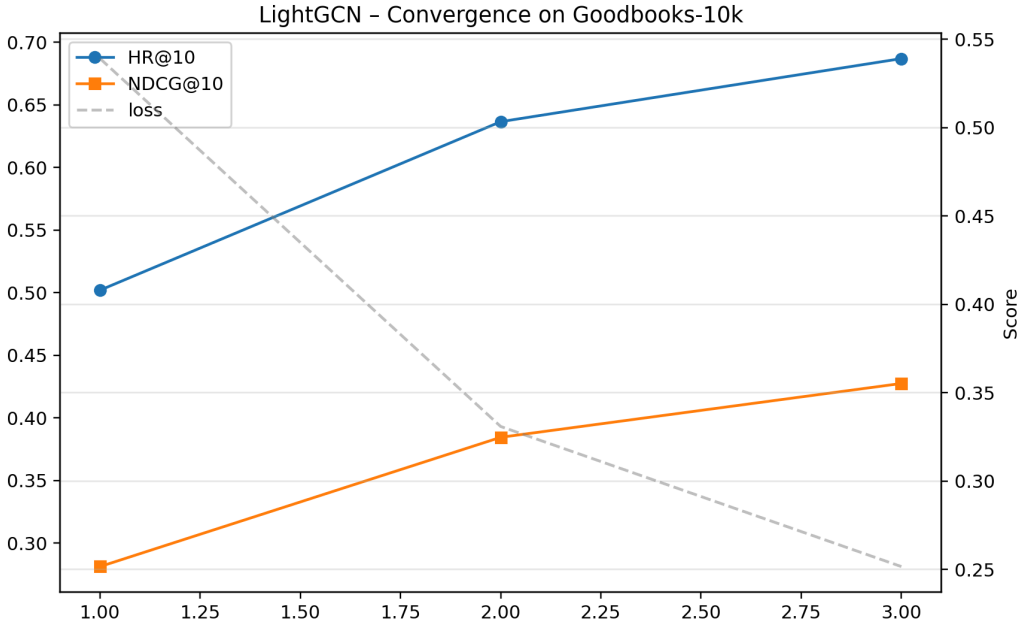

Figure 11. LightGCN training curves.

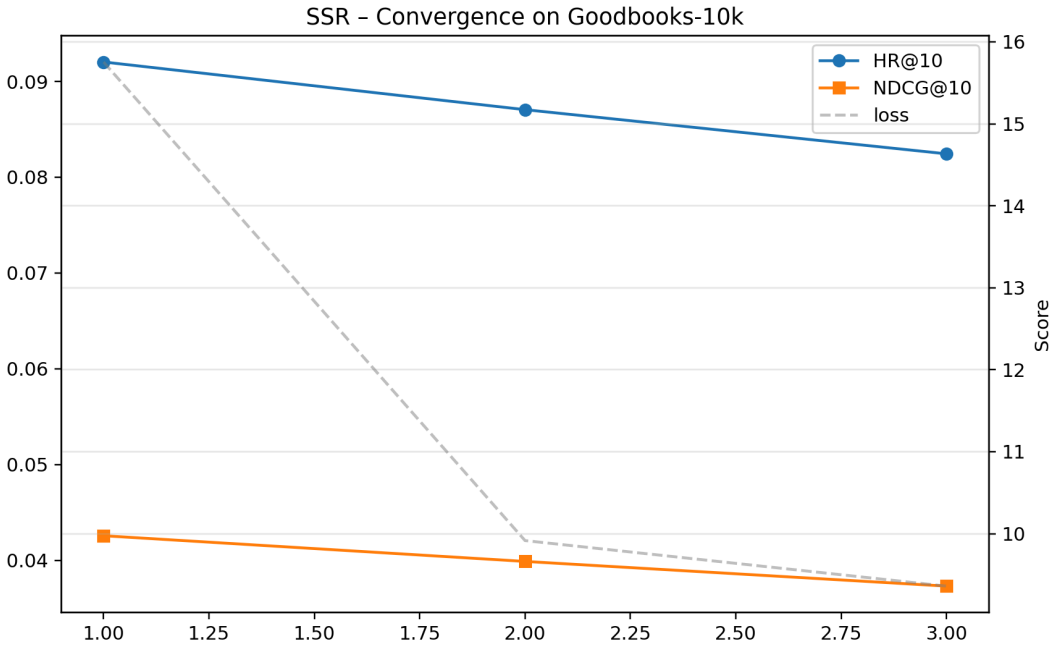

Figure 12. SSR training curves.

Exposure diagnostics: NeuMF/LightGCN: Coverage@10 ~ 1.00 / 0.99, Tail-Coverage@10 ~ 0.38 / 0.31, Gini ~ 0.46 / 0.45, Novelty ~ 13.4 bits, aligned with exposure-aware guidance ([7]-[9], [11], [12], [18], [22]).

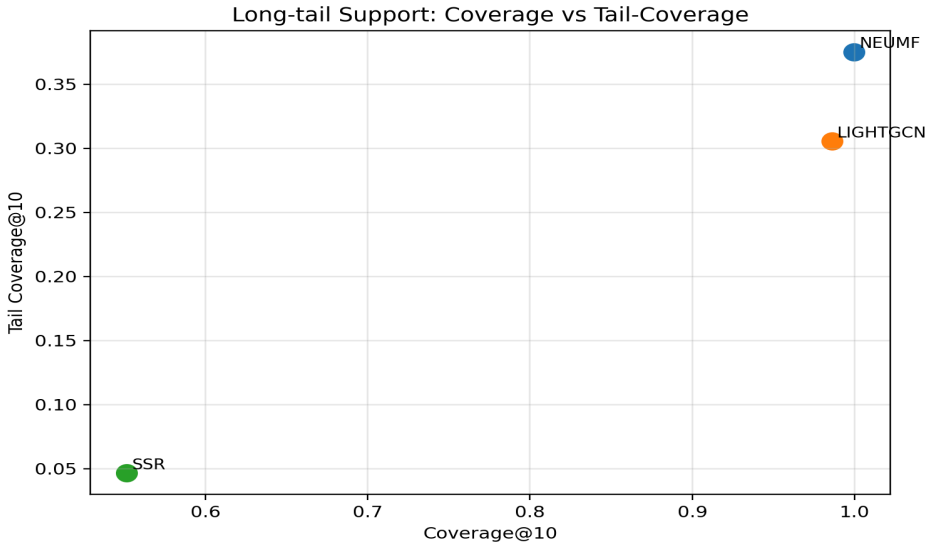

Figure 13. Coverage vs Tail-Coverage at K=10 for each model.

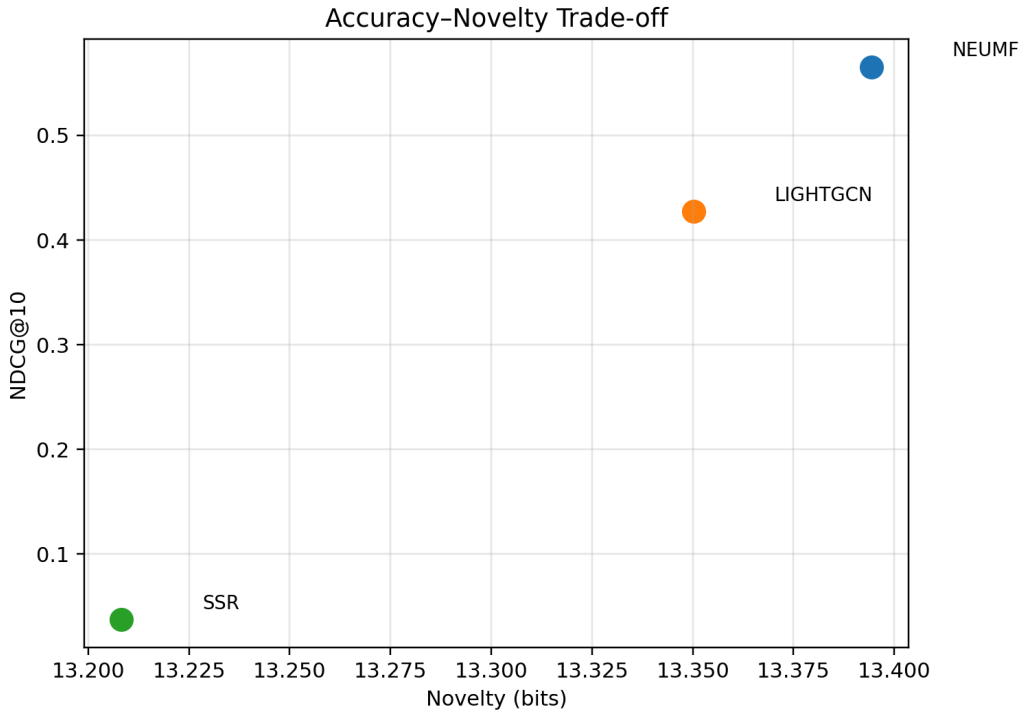

Figure 14. Scatter of Novelty (bits) versus NDCG@10 showing diversity-accuracy trade-off.

Segment analysis: NeuMF leads across sparse, medium, and dense user groups, LightGCN is consistently second, SSR provides no benefit absent temporal order ([3], [4]).

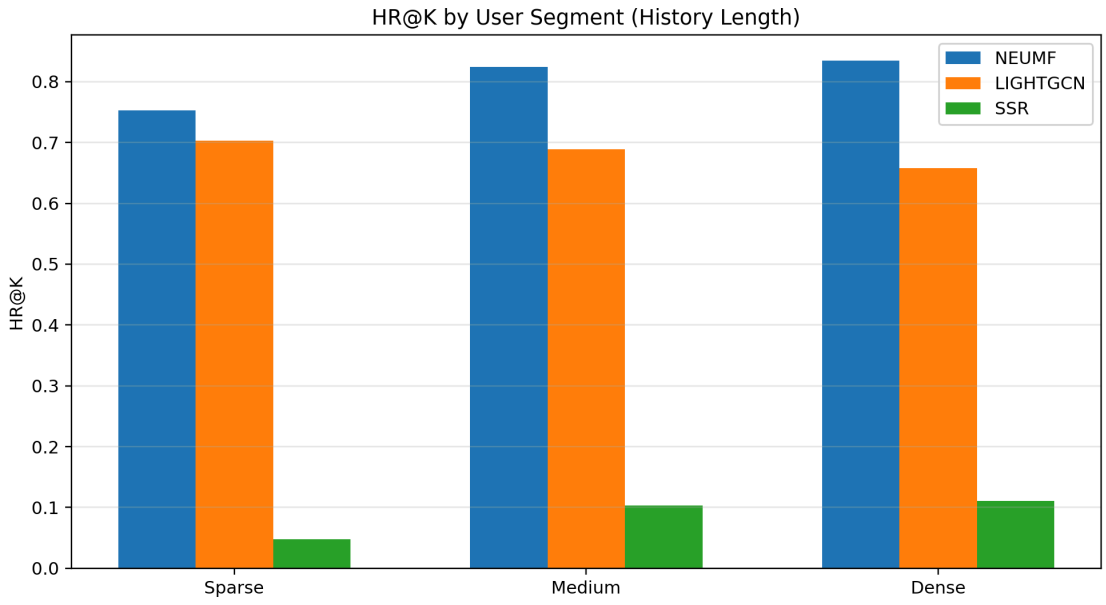

Figure 15. HR@10 by user history segments (sparse, medium, dense).

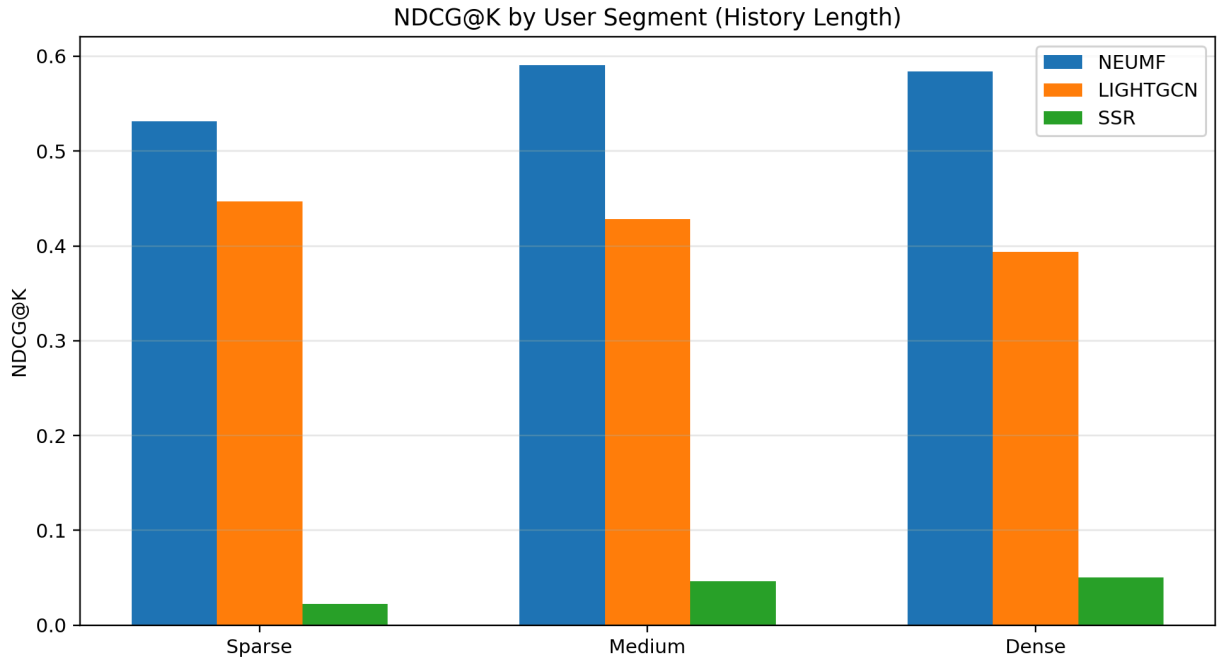

Figure 16. NDCG@10 by user history segments (sparse, medium, dense).

5. Discussion

Interpretation.

- **Data-model fit:** Without temporal order, sequential attention cannot model recency/transition patterns, SSR underperforms as theory predicts ([3], [4]).

- **Why NeuMF wins:** NeuMF's non-linear interaction function fits implicit, non-temporal data and converges rapidly, delivering top-K gains consistent with [1] and fair-evaluation cautions in [19].

- **LightGCN's value:** Graph propagation spreads collaborative signal and enhances catalog breadth (coverage/tail) with a small NDCG trade-off, matching [2], [16]. This makes LightGCN ideal as a reranker/weighted blend with NeuMF.

Implications for researchers.

- Under non-temporal implicit data, neural/graph CF are the baseline of choice, defer sequential models until order exists ([1]-[4], [16], [21]).

- Report exposure metrics (Coverage/Tail-Coverage/Novelty/Gini) with accuracy to reveal bias trade-offs and support calibrated recommendations ([7]-[9], [11], [12], [18], [22]).

- Sampled-metric caveat: keep identical candidates and triangulate with multiple metrics, validate with online A/B when possible ([17], [19], [24]).

Implications for practitioners.

- **Deploy now:** ship NeuMF at K = 10, monitor HR@10, NDCG@10, latency ([1], [20], [23]).

- **Coverage uplift:** add LightGCN as a reranker/blend to lift Tail-Coverage@10, watch NDCG@10 and Gini to avoid over-diversification ([2], [7], [8], [22]).

- **Data plan:** begin timestamp capture, re-benchmark SSR/BERT4Rec when time order exists, consider online A/B if offline gains hold ([3], [4], [20], [23]).

- **Cold-start:** incorporate text metadata and brief onboarding, classic CF guidance and modern industry practice support hybridisation ([11], [13], [14], [20], [21], [23]).

6. Limitations

1. **No timestamps**: SSR cannot learn recency/transition, dataset limitation, not a model flaw ([3], [4]).

2. **Implicit-only labels**: rating >= 3 conflates preference intensity and may reward popularity ([6], [7], [10], [18]).

3. **Sampled metrics**: efficient but biased, we mitigate with identical candidates and multi-metrics, yet online A/B is definitive ([17], [19], [24]).

4. **Limited hyperparameter search**: kept tight for comparability/runtime, broader sweeps could narrow the NeuMF/LightGCN gap ([1], [2], [21]).

5. **Cold-start not addressed**: needs content features and onboarding, standard hybrid approaches apply ([11], [13], [14], [21]).

7. Conclusion and Future Work

Conclusion. On implicit Goodbooks-10k without timestamps, NeuMF is the most suitable top-K recommender (HR@10 = 0.798, NDCG@10 = 0.566) ([1], [19], [21]). LightGCN is a strong alternative that enhances coverage and tail exposure ([2], [16]). SSR should be deferred until temporal order is available ([3], [4]).

Future work / prescriptions.

- Deploy NeuMF now, track HR/NDCG and latency, consider calibration to align with user intent ([18]).
- Blend LightGCN to raise Tail-Coverage@10, monitor NDCG@10 and Gini, explore re-ranking with diversity constraints ([2], [11], [22]).
- Collect timestamps, re-evaluate SSR/BERT4Rec, if offline gains hold, run online A/B ([3], [4], [20], [23]).
- Investigate confidence-weighted objectives ([6]), regularisation ([15]), and hybrid content-CF for cold-start ([11], [13], [14], [21]).

8. Replication Package

How to reproduce: place ratings.csv (Goodbooks-10k) in the correct directory, run the training cell for NeuMF, LightGCN, SSR, then run the figure-generation cells .
Link: https://github.com/deepin-kuchroo/big-data-analysis-and-project-assignment.git

References

[1] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural Collaborative Filtering. *WWW*.

[2] He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. (2020). LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. *SIGIR*.

[3] Kang, W.-C., & McAuley, J. (2018). Self-Attentive Sequential Recommendation (SASRec). *ICDM*.

[4] Sun, F., et al. (2019). BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations. *CIKM*.

[5] Rendle, S. (2009). BPR: Bayesian Personalized Ranking from Implicit Feedback. *UAI*.

[6] Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative Filtering for Implicit Feedback Datasets. *ICDM*.

[7] Steck, H. (2011). Item Popularity and Recommendation Accuracy. *RecSys*.

[8] Abdollahpouri, H., Mansoury, M., Burke, R., & Mobasher, B. (2019). The Unfairness of Popularity Bias in Recommender Systems. *RecSys*.

[9] Jannach, D., Lerche, L., Kamehkhosh, I., & Jugovac, M. (2015). What Recommenders Recommend. *UMUAI*.

[10] Cremonesi, P., Koren, Y., & Turrin, R. (2010). Performance of Recommender Algorithms on Sparse Data. *RecSys*.

[11] Ricci, F., Rokach, L., & Shapira, B. (Eds.). (2011/2022). *Recommender Systems Handbook*. Springer.

[12] Ekstrand, M. D., Burke, R., & Diaz, F. (2022). Fairness and Discrimination in Recommender Systems. arXiv.

[13] Linden, G., Smith, B., & York, J. (2003). Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Computing*.

[14] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. *Computer*.

[15] Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). BPR Revisited. *RecSys*.

[16] Wang, X., He, X., Wang, M., Feng, F., & Chua, T.-S. (2019). Neural Graph Collaborative Filtering. *SIGIR*.

[17] Krichene, W., & Rendle, S. (2020). On Sampled Metrics for Item Recommendation. *KDD*.

[18] Steck, H. (2018). Calibrated Recommendations. *RecSys*.

[19] Dacrema, M. F., Cremonesi, P., & Jannach, D. (2019). Are We Really Making Much Progress? *RecSys*.

[20] Covington, P., Adams, J., & Sargin, E. (2016). Deep Neural Networks for YouTube Recommendations. *RecSys*.

[21] Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep Learning Based Recommender System: A Survey. *IEEE TKDE*.

[22] Vargas, S., & Castells, P. (2011). Rank and Relevance in Novelty and Diversity for Recommender Systems. *RecSys*.

[23] Zhou, Z., et al. (2020). Large-Scale CTR Prediction with Deep Models: Practice & Open Problems. *KDD (Industry)*.

[24] Herlocker, J., Konstan, J., Terveen, L., & Riedl, J. (2004). Evaluating Collaborative Filtering Recommender Systems. *ACM TOIS*.