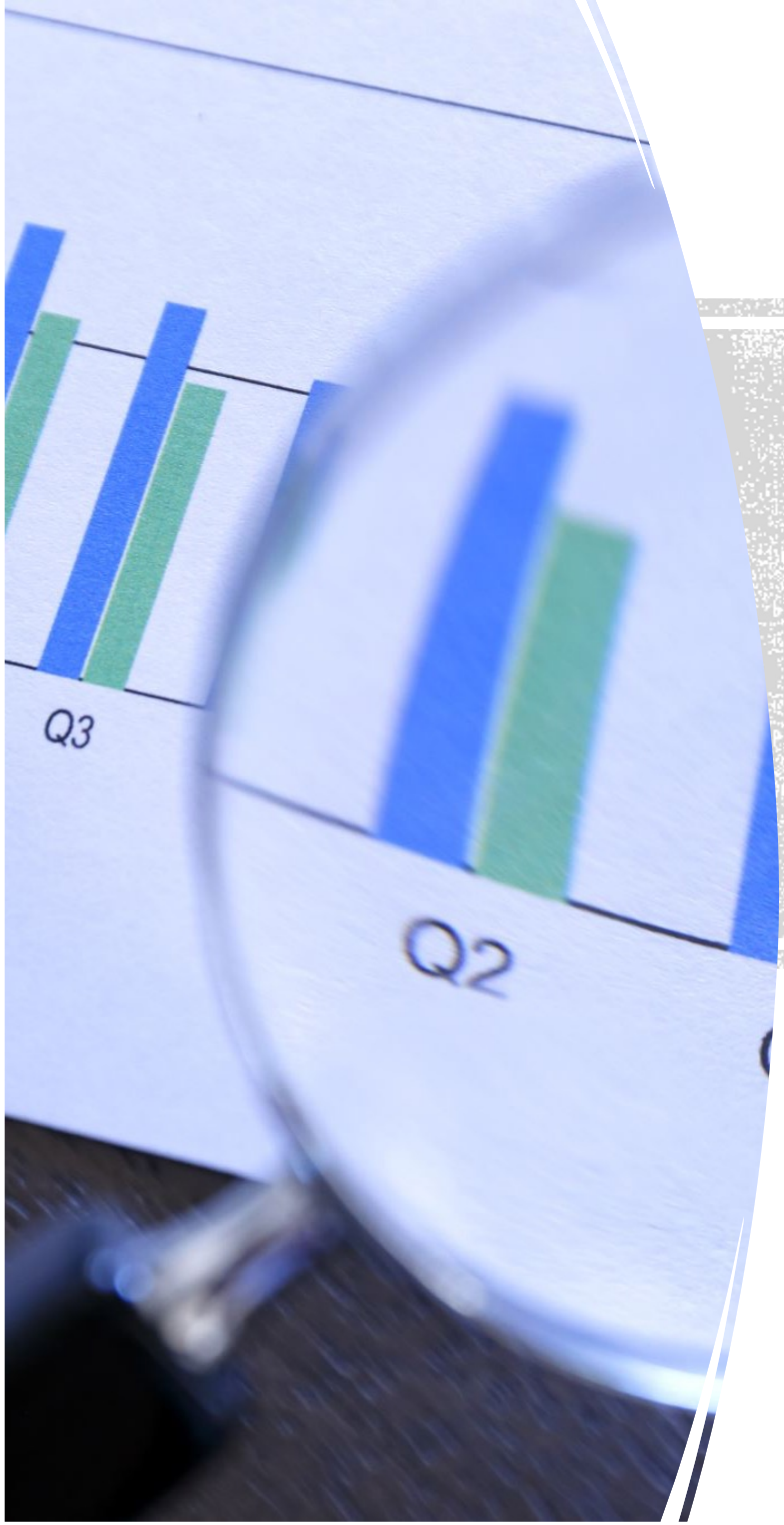


PERFORMANCE ANALYSIS OF RECOMMENDATION METHODS

Assessment 3: Virtual Presentation

- Hishikesh Phukan

S4031214

DECLARATION

I certify that this is all my own original work. If I took any parts from elsewhere, then they were non-essential parts of the assignment, and they are clearly attributed in our submission. I will show I agree to this honor code by typing "Yes":

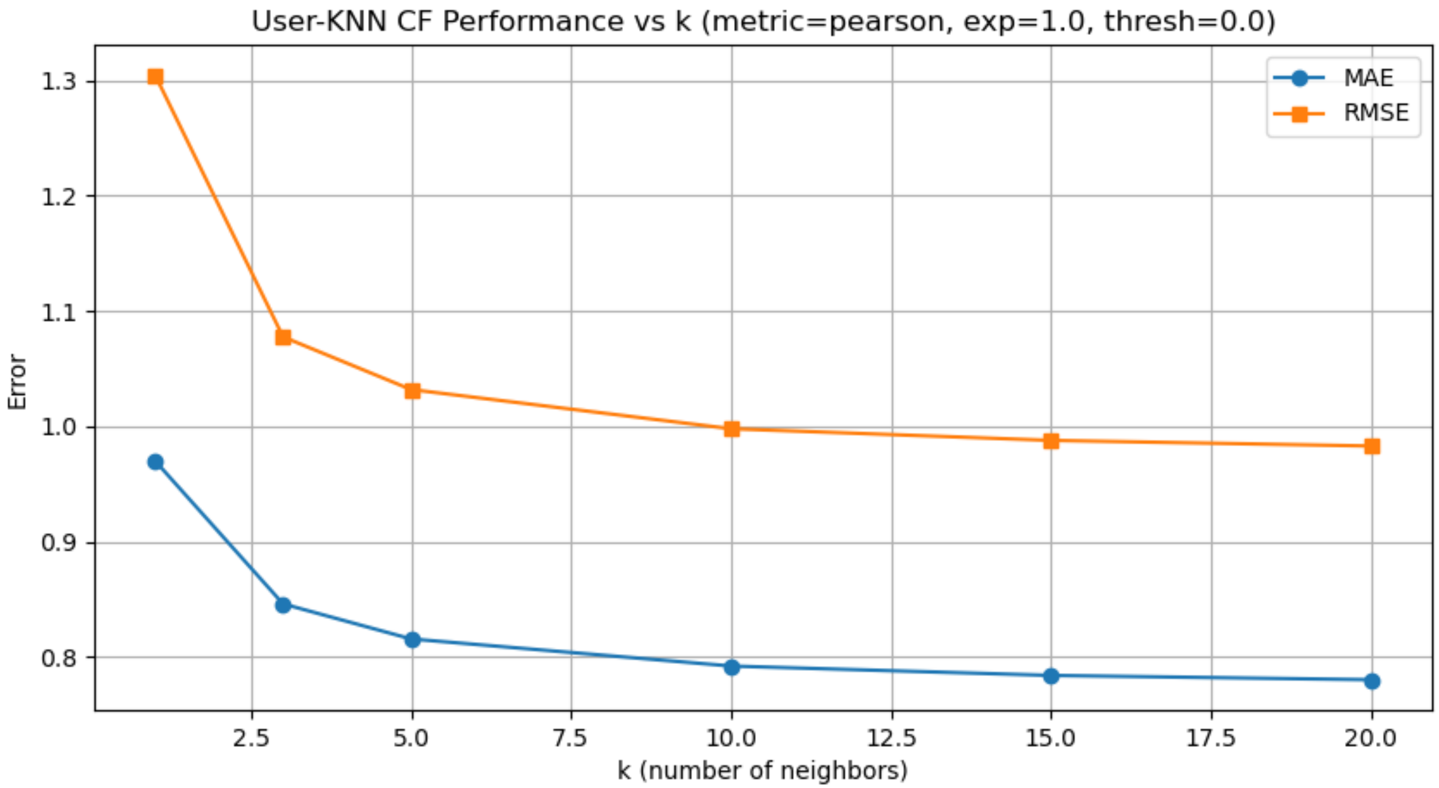
METHOD 3: USER-BASED KNN COLLABORATIVE FILTERING

- Uses cosine and pearson similarity metrics
- Optimizes hyperparameters: k, sim_exponent, sim_threshold
- Falls back to user/global average for missing predictions
- Chosen based on best RMSE and MAE

METHOD 3: USER-BASED KNN COLLABORATIVE FILTERING

```
Metric: pearson | k: 10 | Exponent: 0.5 | Threshold: 0.1 -> MAE: 0.8036, RMSE: 1.0198
Metric: pearson | k: 10 | Exponent: 0.5 | Threshold: 0.2 -> MAE: 0.8412, RMSE: 1.0779
Metric: pearson | k: 10 | Exponent: 1.0 | Threshold: 0.0 -> MAE: 0.7924, RMSE: 0.9980
Metric: pearson | k: 10 | Exponent: 1.0 | Threshold: 0.1 -> MAE: 0.8033, RMSE: 1.0195
Metric: pearson | k: 10 | Exponent: 1.0 | Threshold: 0.2 -> MAE: 0.8415, RMSE: 1.0780
Metric: pearson | k: 15 | Exponent: 0.5 | Threshold: 0.0 -> MAE: 0.7847, RMSE: 0.9884
Metric: pearson | k: 15 | Exponent: 0.5 | Threshold: 0.1 -> MAE: 0.7993, RMSE: 1.0142
Metric: pearson | k: 15 | Exponent: 0.5 | Threshold: 0.2 -> MAE: 0.8413, RMSE: 1.0780
Metric: pearson | k: 15 | Exponent: 1.0 | Threshold: 0.0 -> MAE: 0.7842, RMSE: 0.9879
Metric: pearson | k: 15 | Exponent: 1.0 | Threshold: 0.1 -> MAE: 0.7990, RMSE: 1.0138
Metric: pearson | k: 15 | Exponent: 1.0 | Threshold: 0.2 -> MAE: 0.8416, RMSE: 1.0781
Metric: pearson | k: 20 | Exponent: 0.5 | Threshold: 0.0 -> MAE: 0.7809, RMSE: 0.9834
Metric: pearson | k: 20 | Exponent: 0.5 | Threshold: 0.1 -> MAE: 0.7977, RMSE: 1.0127
Metric: pearson | k: 20 | Exponent: 0.5 | Threshold: 0.2 -> MAE: 0.8414, RMSE: 1.0780
Metric: pearson | k: 20 | Exponent: 1.0 | Threshold: 0.0 -> MAE: 0.7805, RMSE: 0.9830
Metric: pearson | k: 20 | Exponent: 1.0 | Threshold: 0.1 -> MAE: 0.7974, RMSE: 1.0123
Metric: pearson | k: 20 | Exponent: 1.0 | Threshold: 0.2 -> MAE: 0.8416, RMSE: 1.0781
Best Parameters: {'metric': 'pearson', 'k': 20, 'sim_exponent': 1.0, 'sim_threshold': 0.0} -> MAE: 0.7805, RMSE: 0.9830
```

METHOD 3: USER-BASED KNN COLLABORATIVE FILTERING



METHOD 4: ITEM-BASED KNN COLLABORATIVE FILTERING

- Similar approach but with item-item similarities
- Optimizes k and similarity type
- Falls back to item average when necessary
- Chosen based on evaluation metrics

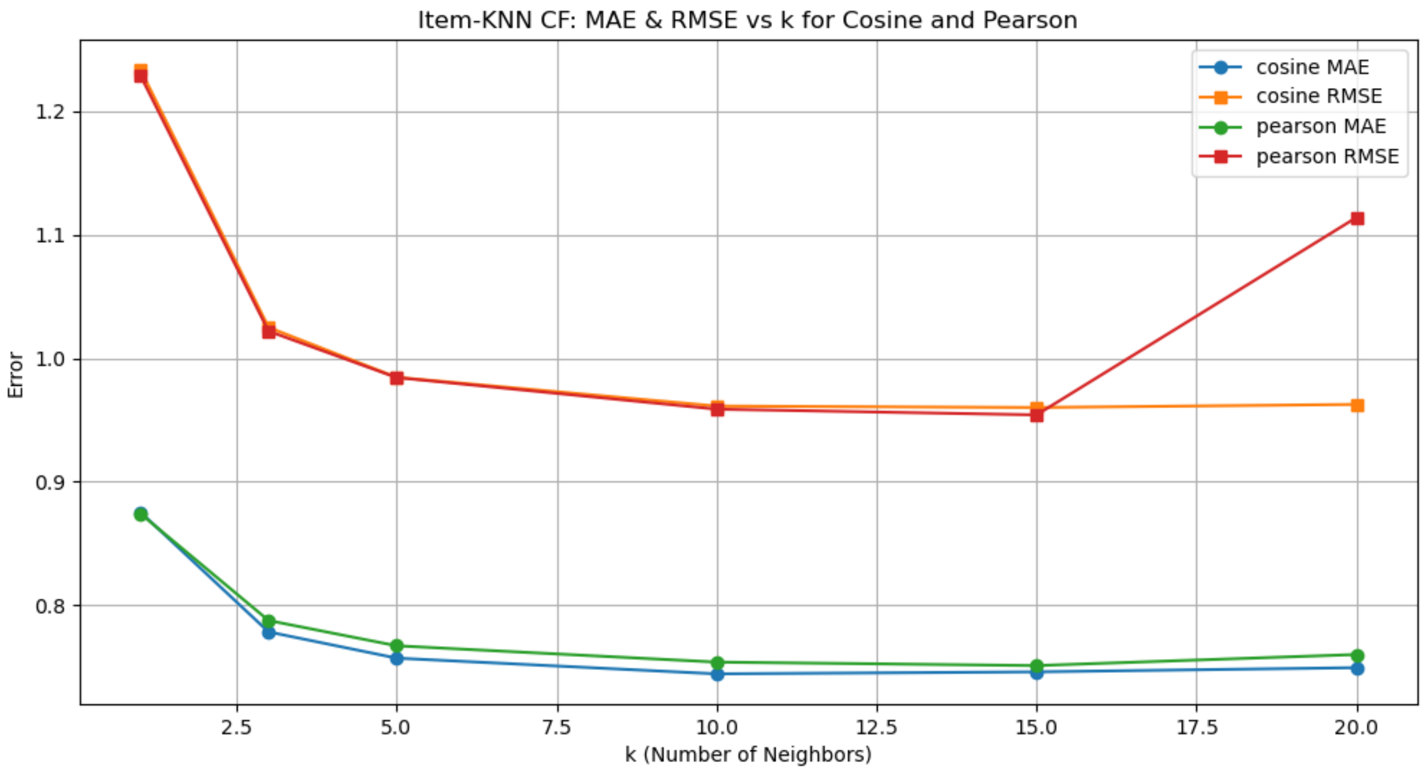
METHOD 4: ITEM-BASED KNN COLLABORATIVE FILTERING

```
All Results for Item-KNN (MAE, RMSE) per similarity metric and k:

Metric: cosine  | k:  1 -> MAE: 0.8752, RMSE: 1.2336
Metric: cosine  | k:  3 -> MAE: 0.7783, RMSE: 1.0249
Metric: cosine  | k:  5 -> MAE: 0.7570, RMSE: 0.9845
Metric: cosine  | k: 10 -> MAE: 0.7443, RMSE: 0.9612
Metric: cosine  | k: 15 -> MAE: 0.7459, RMSE: 0.9600
Metric: cosine  | k: 20 -> MAE: 0.7493, RMSE: 0.9625
Metric: pearson | k:  1 -> MAE: 0.8740, RMSE: 1.2288
Metric: pearson | k:  3 -> MAE: 0.7875, RMSE: 1.0219
Metric: pearson | k:  5 -> MAE: 0.7670, RMSE: 0.9843
Metric: pearson | k: 10 -> MAE: 0.7538, RMSE: 0.9587
Metric: pearson | k: 15 -> MAE: 0.7510, RMSE: 0.9541
Metric: pearson | k: 20 -> MAE: 0.7600, RMSE: 1.1140

Best metric: pearson -> MAE: 0.7510, RMSE: 0.9541
```

METHOD 4: ITEM-BASED KNN COLLABORATIVE FILTERING



METHOD 5: HYBRID USER-ITEM KNN

- Combines Method 3 and 4 predictions
- Uses blending with lambda (0 to 1, step 0.05)
- λ * User-based + (1 - λ) * Item-based prediction
- Optimal λ chosen for lowest RMSE

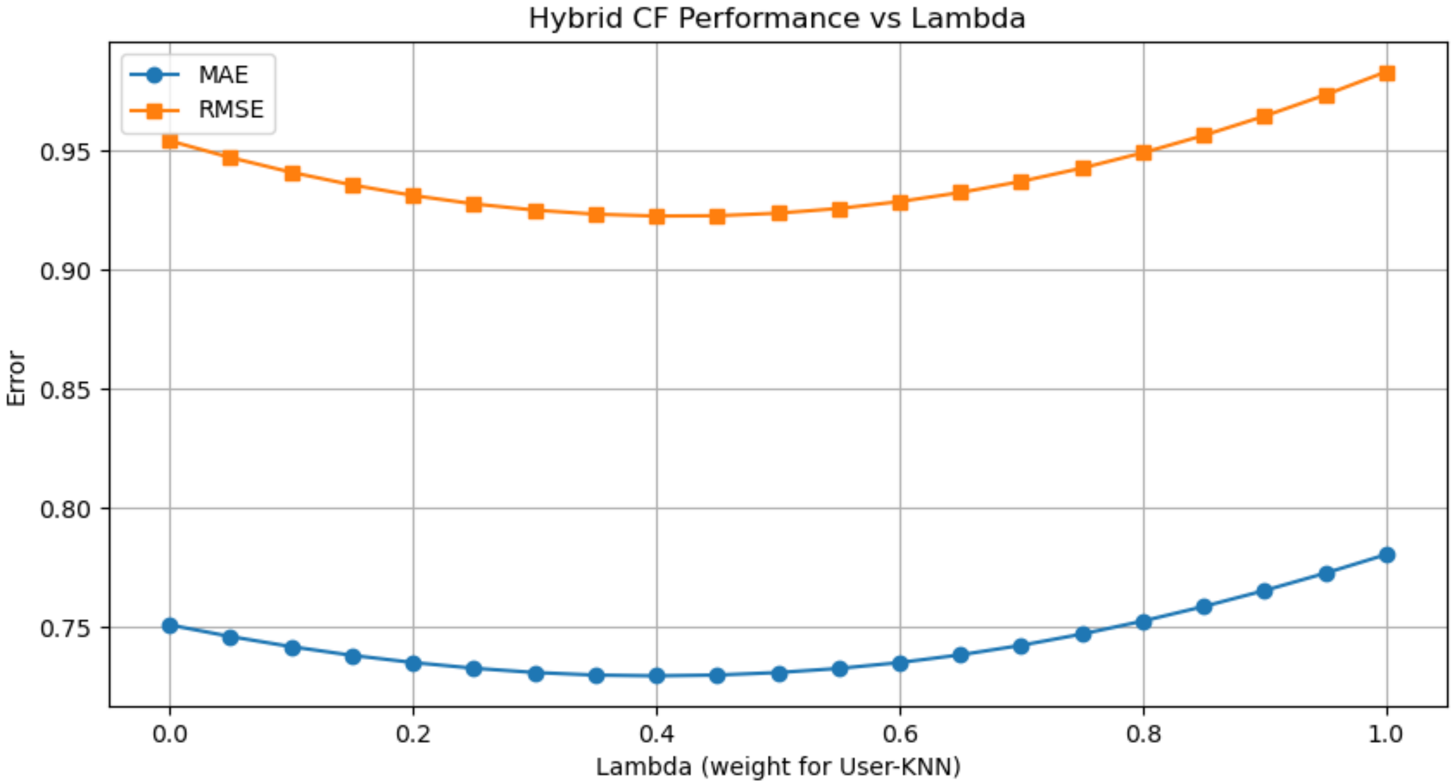
METHOD 5: HYBRID USER-ITEM KNN

```
Hybrid CF Results (MAE, RMSE) for each lambda value:

Lambda: 0.00 -> MAE: 0.7510, RMSE: 0.9541
Lambda: 0.05 -> MAE: 0.7461, RMSE: 0.9470
Lambda: 0.10 -> MAE: 0.7419, RMSE: 0.9409
Lambda: 0.15 -> MAE: 0.7382, RMSE: 0.9356
Lambda: 0.20 -> MAE: 0.7352, RMSE: 0.9311
Lambda: 0.25 -> MAE: 0.7328, RMSE: 0.9276
Lambda: 0.30 -> MAE: 0.7310, RMSE: 0.9250
Lambda: 0.35 -> MAE: 0.7300, RMSE: 0.9233
Lambda: 0.40 -> MAE: 0.7296, RMSE: 0.9225
Lambda: 0.45 -> MAE: 0.7300, RMSE: 0.9226
Lambda: 0.50 -> MAE: 0.7310, RMSE: 0.9237
Lambda: 0.55 -> MAE: 0.7327, RMSE: 0.9257
Lambda: 0.60 -> MAE: 0.7352, RMSE: 0.9286
Lambda: 0.65 -> MAE: 0.7384, RMSE: 0.9324
Lambda: 0.70 -> MAE: 0.7424, RMSE: 0.9370
Lambda: 0.75 -> MAE: 0.7472, RMSE: 0.9426
Lambda: 0.80 -> MAE: 0.7526, RMSE: 0.9490
Lambda: 0.85 -> MAE: 0.7587, RMSE: 0.9563
Lambda: 0.90 -> MAE: 0.7654, RMSE: 0.9644
Lambda: 0.95 -> MAE: 0.7727, RMSE: 0.9733
Lambda: 1.00 -> MAE: 0.7805, RMSE: 0.9830

Best Lambda: 0.40 -> MAE: 0.7296, RMSE: 0.9225
```

METHOD 5: HYBRID USER-ITEM KNN



PERFORMANCE COMPARISON OF METHODS 1–5

Method 1: User average (baseline), least accurate

- MAE: 0.82, RMSE: 1.03

Method 2: Item average, slightly better

- MAE: 0.79, RMSE: 1.001

Method 3 & 4: KNN models, improved accuracy

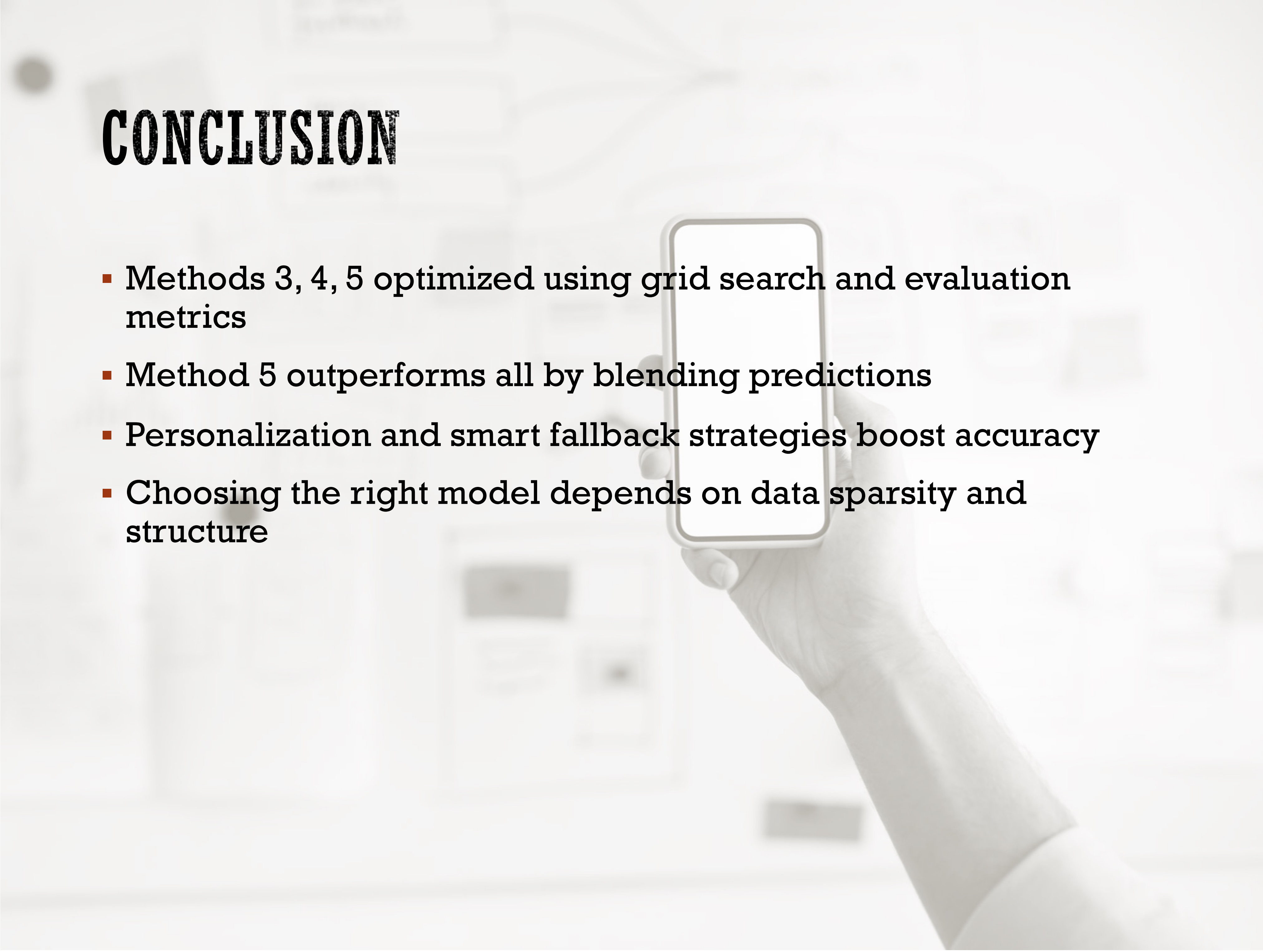
- MAE(User): 0.78, RMSE(User): 0.98
- MAE(Item): 0.75, RMSE(Item): 0.95

Method 5: Hybrid model, best performance overall

- MAE: 0.72, RMSE: 0.92

WHY SOME METHODS PERFORM BETTER

- Baselines lack personalization
- KNN methods account for similar users/items
- Hybrid model leverages strengths of both sides
- Optimal tuning of hyperparameters is key



CONCLUSION

- Methods 3, 4, 5 optimized using grid search and evaluation metrics
- Method 5 outperforms all by blending predictions
- Personalization and smart fallback strategies boost accuracy
- Choosing the right model depends on data sparsity and structure

KEY LEARNINGS AND TAKEAWAYS

Personalization is essential

- Models that adapt to user preferences deliver significantly better accuracy

KNN methods outperform simple baselines

- User and item similarity leads to more relevant recommendations

Hybrid models offer the best of both worlds

- Combining user-based and item-based insights improves robustness

Hyperparameter tuning is critical

- Fine-tuning kkk, similarity metrics, and λ\lambdaλ greatly affects performance

Visualization supports decision-making

- Plots of RMSE/MAE trends made it easier to choose optimal configurations

LIMITATION AND FUTURE WORK

Current Limitation

- Cold-start problem for new users/items with limited data
- KNN scalability is a concern on larger datasets
- Limited use of contextual or content-based features
- Dependence on explicit ratings only

Future Directions

- Explore matrix factorization with better regularization
- Integrate content-based or context-aware recommendations
- Experiment with deep learning models for better scalability and feature learning
- Use side information like tags, timestamps, genres, or user profiles to improve accuracy

THANK YOU ☺

REFERENCES

- Herlocker, J. L., Konstan, J. A., & Riedl, J. (2002). *An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms*. Information Retrieval, **5**(4), 287–310. https://doi.org/10.1023/A:1020443909834
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). *Item-based collaborative filtering recommendation algorithms*. In Proceedings of the 10th international conference on World Wide Web (pp. 285–295). https://doi.org/10.1145/371920.372071
- Ricci, F., Rokach, L., & Shapira, B. (2015). *Recommender systems: Introduction and challenges*. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Recommender systems handbook* (2nd ed., pp. 1–34). Springer. https://doi.org/10.1007/978-1-4899-7637-6_1
- Aggarwal, C. C. (2016). *Neighborhood-based collaborative filtering*. In *Recommender Systems: The Textbook* (pp. 75–125). Springer. https://doi.org/10.1007/978-3-319-29659-3_3
- Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). *Collaborative filtering recommender systems*. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 291–324). Springer. https://doi.org/10.1007/978-3-540-72079-9_9