ISSN: 2834-7706

Shot Communication

Journal of Mathematical Techniques and Computational Mathematics

Machine Learning System Design: Multi-Model-Based Recommendation & Identification

Fanfei Meng*

Department of Electrical and Computer Engineering Northwestern University

***Corresponding Author**
Fanfei Meng, Department of Electrical and Computer Engineering Northwestern University

Submitted: 2025, Feb 13; **Accepted:** 2025, Mar 11; **Published:** 2025, Mar 18

Citation: Meng., F. (2025). Machine Learning System Design: Multi-model-based Recommendation & Identification. *J Math Techniques Comput Math, 4* (2). 01-04.

Abstract

This document presents a comprehensive design framework for two machine learning systems aimed at optimizing recommendation and identification tasks in distinct domains: an Ads Ranking System and a Family-Friendly Listing Ranking System. Both systems leverage multi-modal data, advanced modeling techniques, and robust evaluation methods to achieve high performance and scalability.

The Ads Ranking System prioritizes ads for user engagement and revenue optimization through short-term metrics such as Click-Through Rate (CTR), Conversion Rate (CVR), and Revenue Per Mile (RPM), alongside long-term metrics including user retention and model latency. It integrates diverse data sources, including user behavior, ad content (text, images, tabular data), and contextual information. The system employs feature engineering techniques to generate embeddings for visual, textual, and tabular data and uses models ranging from XGBoost to advanced neural architectures like Deep Interest Networks (DIN). Offline and online evaluation metrics such as AUC, NDCG, and real-time business metrics ensure robust performance monitoring and iterative improvement. The Family-Friendly Listing Ranking System focuses on classifying and ranking listings for family-friendliness, considering features such as amenities, reviews, and location safety. The model strategy incorporates tree-based methods for interpretability and multi-tower neural networks for handling unstructured data. Evaluation involves precision, recall, and ranking metrics alongside A/B testing to align offline improvements with business goals.

Challenges like data distribution shifts and user experience mismatches are addressed through feature refinement and explain ability tools.

This work highlights the integration of machine learning, multi-modal data processing, and systematic evaluation to build scalable and impactful recommendation systems. It also underscores the importance of balancing interpretability, computational efficiency, and long-term user satisfaction.

1. Design an Ads Ranking System (Meta Machine Learning Interview Loop)

1.1 Metrics

1.1.1 Short-Term Metrics

- **CTR** (Click-Through Rate): Measures the percentage of users who click on an ad.
- **CVR** (Conversion Rate): Measures the percentage of users who complete a desired action after clicking.
- **RPM** (Revenue Per Mile): Calculates revenue per 1,000 impressions.

1.1.2 Long-Term Metrics

- **User Retention:** Tracks the percentage of users returning to engage with ads over time.
- **Online CTR/CVR Performance:** Monitors real-time click and conversion rates.
- **Model Latency:** Evaluates the computational efficiency of the model.

1.2 Data Sources

1.2.1 User Data

- **Interaction History:** Sequence of ads interacted with (long-term and short-term).
- **User Profile:** Demographics, preferences, and historical behavior.
- Social Network Data: Connections, friends, or shared activities.

1.2.2 Ad Content

- **Visual Data:** Images or visual media associated with ads.
- **Textual Data:** Ad descriptions, titles, or keywords.
- **Tabular Data:** Features like bid amount, categories, or relevance scores.

1.2.3 Contextual Data

- **Time:** Hour of the day or day of the week.
- **Device:** Type of device used (mobile, desktop).
- **Location:** Geographic data.

1.2.4 Interaction Window

- 7-day sequence of all ads/content the user interacted with: [item1, item2, item3, ..., item_n].

1.3 Feature Engineering

1.3.1 Visual Embeddings

- Train an embedding model to represent visual data compactly.
- Use **similar pairs** (e.g., original vs. augmented samples, items of the same category).
- Use **discriminative pairs** (e.g., dog vs. cat, cute dog vs. dangerous dog, dog vs. wolf).

1.3.2 Text Embeddings

- Represent textual data using embedding models like Transformer-based encoders.

1.3.3 Tabular Embeddings

- Transform tabular data (e.g., bid amounts, categorical features) into embeddings for integration with other modalities.

1.3.4 Labeling Data

- **Positive Labels:** Ads clicked by the user.
- **Negative Labels:** Ads not clicked by the user.
- For ranking, retain the top 30 items from a list of 100 and discard the bottom 70.

1.4 Modeling Approach

1.4.1 Baseline Models:

- **XGBoost:** Predict CTR using engineered features.

1.4.2 Deep Learning Models:

- **DNN:** Combine embeddings for visual, textual, and tabular data into a unified representation for CTR prediction.

1.4.3 Advanced Modeling

- **Deep Interest Network (DIN)**

Incorporates both short-term and long-term user interaction sequences.

Computes softmax attention over target items ([item1, item2, ..., item_n]) and user interactions.

Formula:

softmax[item1_att_target, item2_att_target, ..., item_n_att_target] * [item1, item2, ..., item_n]

1.5 Evaluation

1.5.1 Offline Metrics

- **AUC for CTR:** Measures the model's ability to rank clicked items higher than non-clicked ones.

1.5.2 Online Metrics

- **RPM:** Revenue generation efficiency.
- **User Retention:** Tracks long-term engagement with ads.
- **Online CTR/CVR Performance:** Real-time evaluation of ad effectiveness.

1.5.3 Model Latency

- Ensure fast inference to maintain a seamless user experience.

1.6 Ranking Analysis

1.6.1 Ranking Comparison

- Use the same dataset to generate ranking lists with the new and old models.
- Focus on the top-ranked item to assess the difference between new_rank1 and old_rank1.

1.6.2 Evaluation Metrics

- Rank 1 item click-through and conversion differences.
- Coverage of new versus old items in the ranking list.

2. Family-Friendly Listing Ranking System Design (Airbnb Machine Learning Interview Loop)

Part 1: Problem Formulation and Data Sources

2.1 Objective: Build a system to classify and rank listings for family-friendliness.

- **Classification Problem:** Determine if a listing is family-friendly.
- **Ranking Problem:** Prioritize the most family-friendly listings when the filter is applied.

2.2 Labels

2.2.1 Explicit Labels

- Family-specific amenities (e.g., cribs, high chairs).
- Host-provided information (e.g., "family-friendly" tag).

2.2.1 Implicit Labels

- User reviews (mentions of children, safety, etc.).
- Booking patterns (e.g., booked frequently by families or during school holidays).
- User profiles (past family-oriented booking behavior).

2.3.1 Label Creation

- Use human annotators to label listings as family-friendly.
- Leverage semi-supervised models to expand the labeled dataset.
- Collect user feedback on listings to refine labels over time.

2.3 Data Sources

- Listing Data: Descriptions, amenities, and hosting information.
- Pricing & Availability: Affordability and booking windows.
- User Reviews: Sentiment and relevance to family stays.
- Booking Patterns: Historical data on family bookings.
- User Behavior Data: Click-through rates (CTR), conversion rates, and other actions.

Part 2: Features and Model Strategy

2.4 Features

- **Amenities:** Family-oriented amenities (e.g., high chairs) and safety features (e.g., smoke detectors).
- **Listing Type:** Entire homes with kitchens, dining rooms, etc.
- **Location:** Safe neighborhoods, proximity to parks, and low crime rates.
- **Reviews:** Cleanliness, safety, and mentions of children.
- **Negative Signals:** Adult-only listings or descriptions indicating non-family suitability.

2.5 Model Strategy

2.5.1 Baseline Models

- **Tree-based Models:** XGBoost, logistic regression for interpretability.
- Pros: Simple and interpretable.
- Cons: Inefficient for unstructured data like text.

2.5.2 Advanced Text Embedding:

- Use text encoders (e.g., Word2Vec, Transformers) to convert descriptions and reviews into dense embeddings.
- Combine embeddings with tabular data to enrich feature representation.

2.6 Deep Learning Models

- **Start with simple architectures (e.g., Multi-Layer Perceptron, MLP).**
- Predict family-friendly probabilities and rank listings accordingly.
- Pros: Efficient and potentially more powerful than tree-based models.
- Cons: Text information can be hard to capture and interpret.

2.7 Multi-Tower Neural Network:

- Design specialized towers for different input types:
- User-side text features.
- Host-side text features.
- Numerical/tabular data (amenities, location, pricing).
- Combine outputs using Multi-Head Attention (MHA) for a holistic view.
- Final output predicts the family-friendliness score for ranking.

Part 3: Evaluation

2.8Offline Evaluation

- Metrics for Classification:
- Precision, Recall, F1 Score, AUC.
- Metrics for Ranking:
- NDCG (Normalized Discounted Cumulative Gain), MRR (Mean Reciprocal Rank), Precision@K.

2.9 Online Evaluation

- A/B Testing
- Control group: Current ranking system.
- Treatment group: New ranking system.
- Canary release: Start with 99% control and 1% treatment.
- Measure business metrics (e.g., conversion rates, CTR).
- Gradually ramp up treatment traffic based on performance.

Part 4: Debugging and Explainability

2.10 Issue: Offline metrics improve, but no improvement in business metrics during A/B testing.

2.11 Potential Causes

2.11.1 Model Generalization Issues:

- Offline metrics (e.g., F1 score) may not align with business goals.
- Switch focus to ranking metrics like Precision@K or NDCG.

2.11.2 Production Behavior

- Data or feature distribution shift between training and production.
- Compare training and test distributions to identify discrepancies.
- Analyze feature importance and distributions for anomalies.

2.11.3 User Experience Mismatch:

- Model may over-prioritize certain features (e.g., affordability) that misalign with user expectations.

2.12 Solutions

- Regularly monitor production data to detect shifts.
- Improve feature engineering to capture business-relevant patterns.
- Add explainability tools to debug why the model ranks certain listings higher or lower.

References

1. Meng, F., & Wang, Y. (2023). Transformers: Statistical interpretation, architectures and applications. *Authorea Preprints*.
2. Meng, F. (2023). Research on text recognition methods based on artificial intelligence and machine learning. *Advances in Computer and Communication, 4*(5)..
3. Meng, F., & Wang, C. A. (2023). Sentiment analysis with adaptive multi-head attention in Transformer. *arXiv preprint arXiv:2310.14505*.
4. Razeghi, M., Dehzangi, A., Wu, D., McClintock, R., Zhang, Y., Durlin, Q., ... & Meng, F. (2019, May). Antimonite-based

gap-engineered type-II superlattice materials grown by MBE and MOCVD for the third generation of infrared imagers. *In Infrared Technology and Applications XLV (Vol. 11002, pp. 108-125). SPIE*.

5. Meng, F., Zhang, L., & Wang, Y. (2024). FedEmb: A Vertical and Hybrid Federated Learning Algorithm using Network And Feature Embedding Aggregation. *Proceedings on Engineering*, 6(2), 601-612.
6. Meng, F., Zhang, L., Chen, Y., & Wang, Y. (2023). Sample-Based Dynamic Hierarchical Trans-Former with Layer and Head Flexibility Via Contextual Bandit. *Proceedings on Engineering Sciences*
7. Meng, F., & Wang, C. A. (2024). A Dynamic Interactive Learning Interface for Computer Science Education: Programming Decomposition Tool. *Lecture Notes in Education Psychology and Public Media, 33*, 82-87.
8. Ling, C., Zhang, C., Wang, M., Meng, F., Du, L., & Yuan, X. (2020). Fast structured illumination microscopy via deep learning. *Photonics Research, 8*(8), 1350-1359.
9. Meng, F., Jagadeesan, L., & Thottan, M. (2021). Model-based reinforcement learning for service mesh fault resiliency in a web application-level. *arXiv preprint arXiv:2110.13621*.
10. Wang, Y., Meng, F., Wang, X., & Xie, C. (2023). Optimizing the passenger flow for airport security check. *arXiv preprint arXiv:2312.05259*.
11. Chen, J. J., Xu, Q., Wang, T., Meng, F. F., Li, Z. W., Fang, L. Q., & Liu, W. (2022). A dataset of diversity and distribution of rodents and shrews in China. *Scientific Data, 9*(1), 304.
12. Meng, F., Wang, Y., Zhang, L., & Zhao, Y. (2023). Joint detection algorithm for multiple cognitive users in spectrum sensing. *arXiv preprint arXiv:2311.18599*.
13. Meng, F., & Wang, C. (2024). Artificial Intelligence and Machine Learning Approaches to Text Recognition: A Research Overview. *J. Math. Tech. Comput. Math*, 3, 1-5.
14. Jang, J., Klabjan, D., Mendiratta, V., & Meng, F. (2024). Hybrid FedGraph: An efficient hybrid federated learning algorithm using graph convolutional neural network. *arXiv preprint arXiv:2404.09443*.