Report: MuseLink - A Multi Method Approach to Hybrid Recommendations

Dattatreya Kantha | GWID: G35579113

1. Problem Statement: Personalized music discovery poses challenges due to the complexity and diversity of user preferences, artist characteristics, and interaction networks. Traditional systems often fall short in addressing these aspects, resulting in less engaging recommendations.

This project addresses this by integrating multiple advanced techniques into a hybrid recommendation system for artist and tag-based suggestions:

- **Neural Collaborative Filtering (NCF):** Learns latent user-artist interaction features using deep learning.
- **Graph Neural Networks (GCNs):** Exploits graph structures in user-artist and tag relationships.
- **Content-Based Filtering:** Aligns recommendations with user preferences using artist metadata and tags.
- **Node2Vec Embeddings:** Captures graph properties in interaction data.
- **Hybrid Model Integration:** Combines outputs of all models for robust recommendations.
  This approach ensures comprehensive recommendations by incorporating user behavior, contextual artist data, and graph-based relationships.

1.2 ***Understanding the Domain and Data***

Domain Overview: Music recommendation systems enhance user experiences by integrating interactional, social, and content-based information. This project uses the HetRec 2011 Last.fm dataset, which includes user-artist interactions, tags, and social connections, enabling collaborative, content-based, and graph-based techniques for diverse and personalized recommendations.

Dataset Description

- **Users:** 1,892 users with an average of 49.07 artist interactions and 98.56 tag assignments each.
- **Artists:** 17,632 artists tagged with an average of 8.76 unique tags, providing metadata like name and URL.
- **Tags:** 11,946 tags reflecting genres, moods, and themes.
- **User-Artist Interactions:** 92,834 listening records used for implicit feedback.
- **Social Connections:** 12,717 user friendships for social-based recommendations.
- **Tag Assignments:** 186,479 dynamic tag assignments capturing user preferences over time.

Challenges and Opportunities

- **Data Heterogeneity:** Robust feature engineering integrates interaction, tagging, and social data.
- **Data Sparsity:** Graph-based methods like Node2Vec and GCNs address sparse interactions.
- **Tag Redundancy:** Preprocessing resolves inconsistencies for meaningful filtering.
- **Cold Start Problem:** Tags and social connections enable alternative paths for recommendations.
- **Temporal Dynamics:** Timestamps model shifts in user preferences, allowing adaptive recommendations.

2. Data Cleaning and Handling Missing Values: The dataset underwent a thorough cleaning process to ensure consistency and integrity:

- **Duplicate Removal**: Removed duplicates across all datasets to avoid redundancy.
- **Handling Missing Values**: Addressed 444 missing values in the pictureURL column of the artists dataset by replacing them with 'NoURL'; no missing values were found in other datasets.

- **Normalization**: Listening counts in the user_artists dataset were normalized to a [0,1] range using Min-Max Scaling for consistent model input.
- **Merging Datasets**: Integrated user-artist data with artist metadata and user-tagged data with tags for enriched features.

This process ensured a clean and complete dataset, ready for feature engineering and model development.

2.1 ***Feature Selection and Feature Engineering:*** *Feature* engineering created diverse and meaningful inputs for collaborative, content-based, and hybrid models:

- **User Features**:
    - Built user-artist interaction matrices from normalized listening counts.
    - Generated user-specific tag profiles by aggregating frequently used tags.
- **Artist Features**:
    - Vectorized artist tags using TF-IDF, producing feature-rich representations.
    - Computed artist popularity scores as the sum of normalized interaction weights across users.
- **Graph Features**:
    - Created interaction and tag-based similarity graphs using user-artist relationships.
    - Applied Node2Vec to generate embeddings capturing graph relationships.

2.2 ***Evaluation of Engineered Features****:*

- **Content-Based Similarity**: Cosine similarity between TF-IDF vectors aligned well with artist metadata for accurate recommendations.
- **Graph Embeddings**: Node2Vec embeddings provided robust representations for users and artists.
- **Hybrid Features**: Integrated collaborative, content-based, and graph-based features resulted in a comprehensive and effective feature set.

3. Choice of Appropriate Model(s) : This project employs a combination of collaborative filtering, content-based filtering, graph-based models, and hybrid approaches to address diverse recommendation challenges:

- **Neural Collaborative Filtering (NCF):** Captures latent user-artist interactions using deep learning, suitable for learning complex user preferences.
- **Content-Based Filtering:** Leverages artist metadata and TF-IDF embeddings to recommend similar artists, mitigating the cold-start problem.
- **Graph Convolutional Networks (GCNs):** Models user-artist graphs to capture higher-order relationships, enhancing diversity and accuracy.
- **Node2Vec Embeddings:** Generates low-dimensional graph embeddings for users and artists, providing robust representations.
- **Hybrid Model Integration:** Combines NCF, content-based filtering, and GCN outputs for comprehensive recommendations.

These models were chosen for their ability to handle sparsity, model user behavior, and improve accuracy across diverse scenarios.

3.1 Implementation of Model(s)

- **NCF:**
    - Framework: Implemented in PyTorch with GPU acceleration.
    - Key Components: User and artist embeddings, dense layers, and a sigmoid output layer.
    - Optimization: Binary cross-entropy loss and Adam optimizer.
    - Highlights: Batch normalization and early stopping to prevent overfitting.

- **Content-Based Filtering:**
    - Used TF-IDF to encode artist metadata.
    - Cosine similarity calculated pairwise artist relevance for user recommendations.
- **GCNs:**
    - Framework: Built with PyTorch Geometric.
    - Graph: User-artist interaction graph with normalized listening weights.
    - Features: Combined Node2Vec embeddings with edge weights.
    - Optimization: Binary cross-entropy loss and SGD optimizer.
- **Node2Vec:**
    - Framework: Used Node2Vec library.
    - Parameters: Walk length of 20, 10 walks per node, and 128-dimensional embeddings.
    - Embeddings integrated with GCN and hybrid models.
- **Hybrid Model:**
    - Fusion: Aggregated NCF, content-based, and GCN outputs using weighted fusion.
    - Highlights: Final recommendations incorporated user preferences, artist metadata, and graph relationships for superior accuracy.

4. Evaluation and Performance Metrics

4.1 Appropriate Choice of Evaluation Metrics: *The* following metrics were selected for their relevance to recommendation quality and ranking:

- **Precision@K**: Assesses the proportion of relevant items in the top K results.
- **Recall@K**: Evaluates how many relevant items are retrieved out of all available.
- **NDCG@K**: Measures ranking quality by emphasizing higher-ranked relevant items.
- **Mean Percentage Ranking (MPR)**: Indicates the average rank of relevant items, where lower values signify better performance.
- **Accuracy**: Helps identify model performance and potential overfitting or underfitting.

These metrics provided a comprehensive evaluation framework to analyze recommendation accuracy, diversity, and relevance.

4.2 Performance Comparison of Recommendation Models

Model	Training(%)	Validation (%)	Testing(%)	Precision@10	Recall @10	NDCG @10	MPR	Observations
Baseline	-	-	-	0.1	0.02	0.14	0.2	Generic recommendations with low relevance; lacks personalization.
NCF	93.69	90.65	90.65	0.38	0.9	0.92	0.34	High relevance and precision; struggles with diversity in niche recommendations.
GCN	97.09	97.0	96.54	0.7	0.14	0.81	0.57	Captures graph relationships well; recall is limited for niche recommendations.
Node2Vec	99.0	95.79	95.88	0.93	0.99	0.95	25.27	Strong ranking quality but slightly overfits; .

Content-Based	90.03	86.8	86.95	0.27	0.37	0.54	0.05	Effective for metadata-rich users; struggles with sparsity in data.
Hybrid	98.89	98.88	98.87	0.75	0.9	0.97	0.35	Combines strengths of all models; delivers the best overall performance.

4.3 Baseline vs. Advanced Models

- **Baseline Model**: Relied on popularity-based recommendations with low Precision@10, Recall@10, and NDCG@10, highlighting its limitations in personalization.

- **Advanced Models**: Outperformed the baseline significantly, with the hybrid model excelling due to its integration of collaborative, content-based, and graph-based features.

4.4 Analysis of Recommendations

The analysis highlights the effectiveness and limitations of each model in addressing the recommendation system's goals, as reflected in the evaluation metrics:

- **Baseline Model:** Offers generic, popularity-based recommendations, resulting in low **Precision@10** and **Recall@10**. This underscores the necessity of personalization to improve user engagement and satisfaction

- **Neural Collaborative Filtering (NCF):** Excels in relevance, achieving a high **NDCG@10**, which indicates its capability to rank relevant items effectively. However, its focus on popular items limits diversity in recommendations, suggesting a need for mechanisms to include lesser-known or niche items.

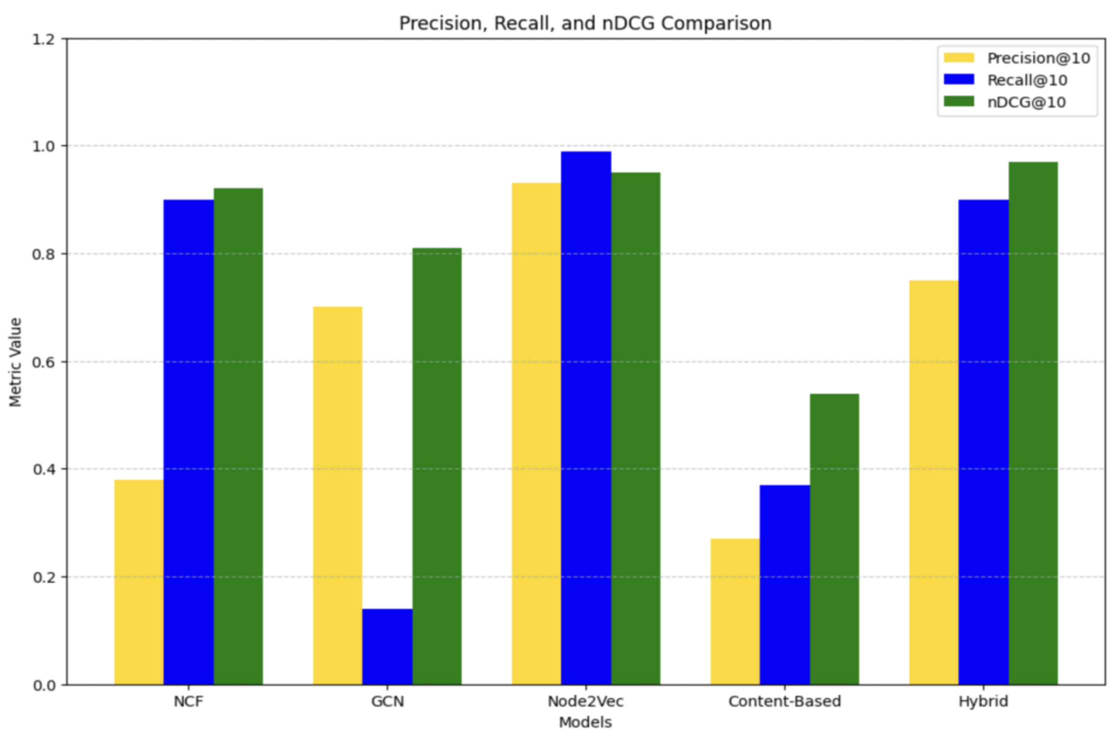
- **Content-Based Filtering:** Performs well for users with detailed metadata and rich tag information, reflected in a reasonable **Precision@10**. However, sparse data for some users results in reduced **Recall@10**, indicating a challenge in covering all relevant recommendations

- **Graph Convolutional Networks (GCNs):** Captures intricate graph relationships, yielding strong **NDCG@10**, showcasing its ability to recommend well-ranked items. However, its lower **Recall@10** highlights a limitation in retrieving a broad range of recommendations, particularly for niche users.

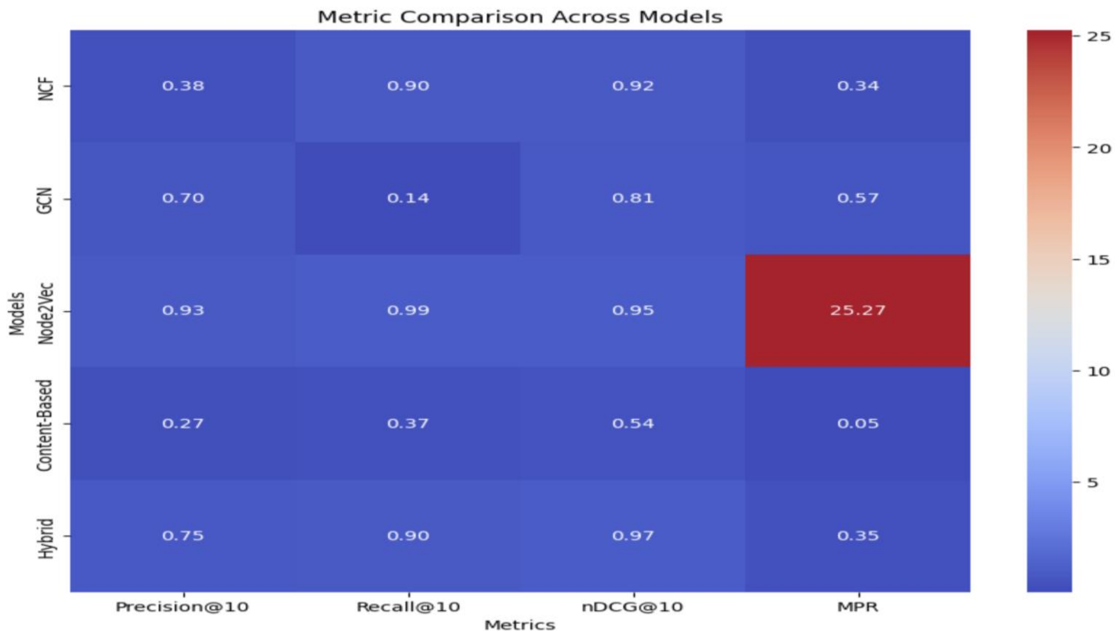
- **Node2Vec:** Offers exceptional ranking quality with high **NDCG@10** and **Recall@10**, demonstrating its strength in both relevance and coverage. Despite its strong performance, slight overfitting is noted, as the model's training and testing accuracies are very close, suggesting room for regularization or additional tuning.

- **Hybrid Model:** Integrates the strengths of all methodologies, achieving the highest scores across **Precision@10**, **Recall@10**, and **NDCG@10**, reflecting its comprehensive approach. The **confusion matrix** further supports this by showing balanced true positive and true negative rates, indicating accurate classification and robust performance across diverse user preferences.

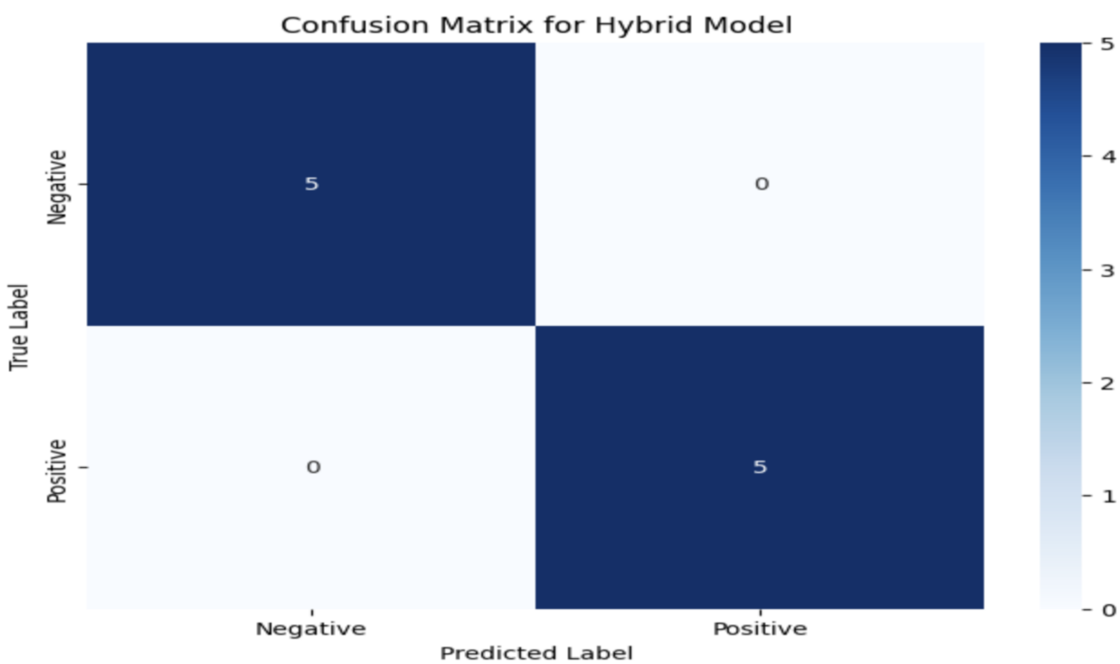
Training, Validation, and Testing Accuracy: Shows Accuracy across datasets, with the hybrid model achieving the best results.



Precision, Recall, and NDCG Comparison: Highlights the hybrid model's superior ranking and coverage metrics.



Metric Comparison Across Models (Heatmap): Compares key performance metrics, showcasing the hybrid model's overall dominance.



Confusion Matrix for Hybrid Model: Validates the hybrid model's accuracy with no misclassifications in the example shown.

4.5 Suggestions for Improvement

- **NCF**: Enhance diversity by promoting less popular or niche recommendations through diversity-focused loss functions.
- **Content-Based Filtering**: Address sparsity by integrating language model embeddings (e.g., BERT) for richer feature representations.
- **GCN**: Add attention mechanisms to focus on the most relevant graph relationships and improve recall for niche recommendations.
- **Node2Vec**: Regularize embeddings to minimize overfitting and enhance its standalone performance.
- **Hybrid Model**: Further optimize weight tuning for model integration and validate performance in real-world scenarios for scalability.

5. Challenges and Conclusion: The project faced challenges like data sparsity, balancing model complexity, ensuring diversity in niche recommendations, and integrating multiple models into a hybrid system. Careful tuning and thorough validation were essential to address these issues.

This project successfully developed a personalized recommendation system using a hybrid approach, integrating collaborative filtering, content-based filtering, and graph-based techniques to achieve superior performance in metrics like Precision@10, Recall@10, and NDCG@10. By addressing challenges such as data sparsity and diverse user preferences, the system delivered balanced and accurate recommendations. The hybrid model outperformed baseline and individual models, highlighting the effectiveness of combining methodologies to capture user behavior, artist attributes, and interaction structures. This work demonstrates the potential of advanced machine learning techniques in building robust recommendation systems and offers a foundation for future enhancements, including transformer-based embeddings and real-time scalability.