SPOTIFY MUSIC RECOMMENDATION SYSTEMS

Low Code Project – MIT Course

Aída Fuentes

August 2025

PROBLEM DEFINITION

Overwhelming choice: Millions of songs make it hard for users to discover music they truly enjoy

Generic rankings fall short: “Top charts” reflect popularity but ignore individual preferences

Need for personalization: A system that learns from user–song interactions to predict what each listener will play next → boosting engagement, retention, and discovery

SOLUTION APPROACH

Challenge of Choice

Users face overwhelming options on streaming platforms, making music discovery difficult despite millions of available songs

Personalized Recommendation

The project aims to develop a system that suggests the top 5 songs for each user, enhancing engagement and reducing search effort through tailored recommendations

Model Evaluation and Adaptability

Approaches like collaborative filtering and content-based techniques are evaluated for accuracy, scalability, and challenges such as sparse data and cold starts

DATA INSIGHTS

37.36

Average Songs per User

209.37

Average Users per Song

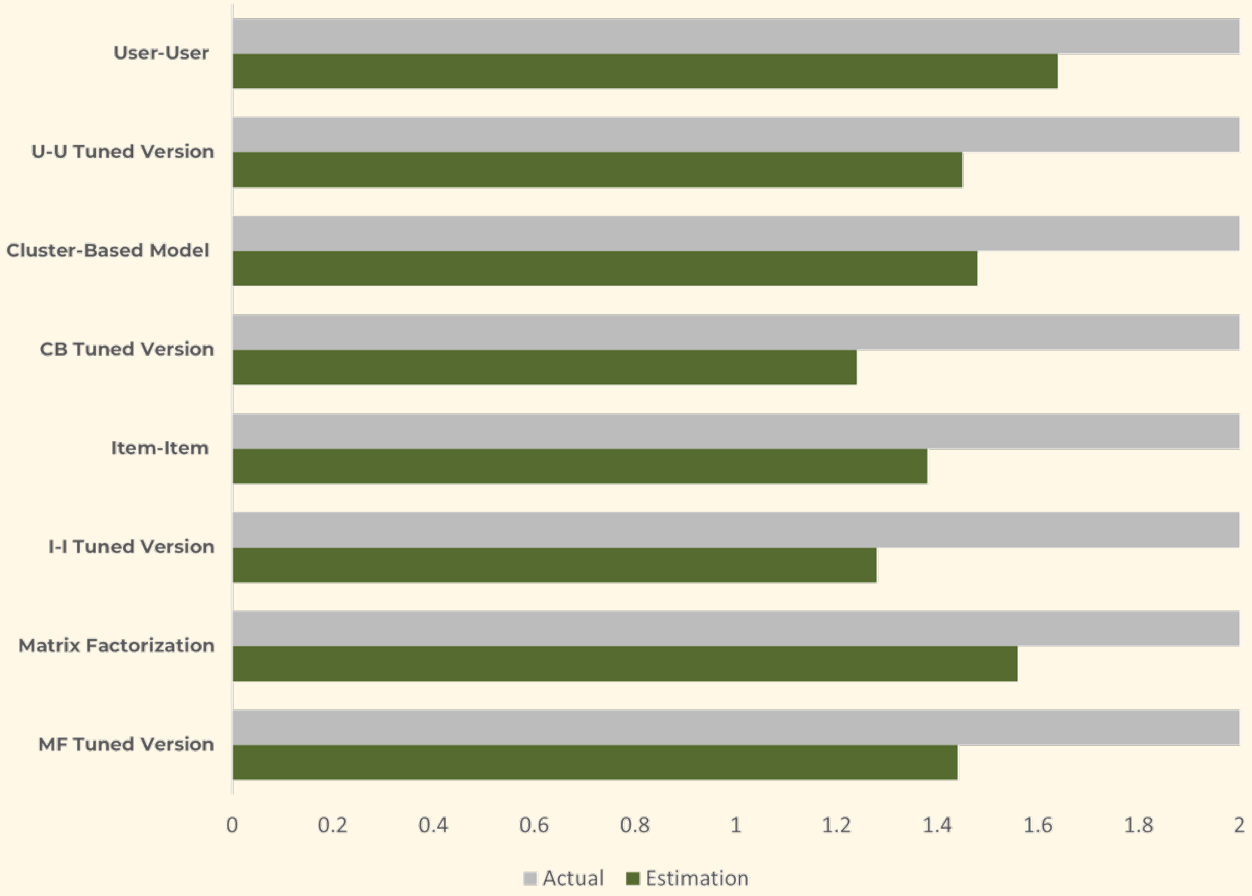
243

Songs by Top User

PROPOSED MODEL SOLUTIONS

- User-User Collaborative Filtering
- Item-Item Collaborative Filtering
- Matrix Factorization (SVD)
- Rank Based Model
- Cluster-Based Model
- Content-Based Model

FINAL MODEL SOLUTION



User-User Collaborative Filtering

The tuned User-User Collaborative Filtering model offers highly personalized and relevant recommendations by identifying users with similar listening histories, demonstrating strong predictive power and consistency across key evaluation metrics

Matrix Factorization (SVD)

Matrix Factorization using SVD performs best in terms of RMSE and scalability, capturing latent features for deeper personalization, making it especially effective in sparse data environments

BUSINESS IMPACT

- **Improved User Engagement:** By recommending songs that closely match user preferences, users are more likely to stay on the platform longer and explore more content

- **Higher Retention:** Personalized recommendations reduce churn by keeping the experience relevant and satisfying

- **Reduced Search Effort:** The system minimizes the time users spend searching, delivering curated experiences that feel tailored and effortless

RECOMMENDATIONS

BENEFITS

- Increased user engagement and time spent on the platform
- Higher retention through better personalization
- Reduced churn and customer acquisition cost over time

COSTS

- Engineering and computational resources to deploy and maintain the models
- Periodic retraining and tuning of collaborative and matrix factorization components

RISKS AND CHALLENGES

RISK

- **Cold Start Problem**
  New users or songs lack data
- **Popularity Bias**
  Over-recommends mainstream hits
- **Scalability**
  High cost with large datasets
- **Model Maintenance**
  Needs constant retraining

MITIGATION

- Combine with content-based models (use metadata like artist, genre, audio features)
- Apply diversity metrics and promote long-tail content
- Use matrix factorization (SVD) and scalable infra (parallelization, cloud)
- Schedule periodic tuning + integrate real-time feedback (likes, skips)

EXECUTIVE SUMMARY

Goal: Build a music recommendation system to deliver **personalized top-5 song suggestions** per user

Approach: Tested multiple models → Rank-Based, User-User, Item-Item, Matrix Factorization (SVD), Clustering, Content-Based

Key Results:

- **User-User (Tuned)** → Best top-N recommendations (highest F1 = 0.525)
- **SVD (Tuned)** → Most accurate predictions (lowest RMSE = 1.0141)

Recommendation: Hybrid solution → combine **User-User** (personalization) + **SVD** (scalability & accuracy)

FURTHER ANALYSIS

- **Cold Start Solutions**: Enrich with metadata & demographics
- **Hybrid Models**: Combine collaborative + content-based
- **Real-Time Feedback**: Skips, likes, replays to refine recs
- **Bias & Fairness**: Monitor popularity/genre bias
- **Playlist-Level Recs**: Personalize entire playlists, not just songs
- **Real-Time Systems**: Explore online learning for low latency
- **Theory vs Practice**: Item-Item may work better in real Spotify (play counts as strong signals) but dataset preprocessing limited results → shows importance of **data richness**

THANK YOU

APPENDIX

Model	Prediction 1 (Known play_count r_ui = 2)		Prediction 2 (Unkown play_count)	
	Estimation	Actual k	Estimation	Actual K
User User Collaborative Filtering	1.8	40	1.64	40
Tuned Version	1.96	24	1.45	10
Item-Item Collaborative Filtering	1.36	20	1.38	20
Tuned Version	1.7	NA	1.28	10
Matrix Factorization (SVD)	1.27	10	1.56	10
Tuned Version	1.54	10	1.44	10
Cluster-Based Model	1.29	10	1.48	10
Tuned Version	1.91	10	1.24	10

APPENDIX

```
# Make top 5 recommendations for user_id 6958 with a similarity-based recommendation engine
#get_recommendations(df_final, "A3LDPF5FMB782Z", 5, sim_user_user_optimized)
recommendations = get_recommendations(df_final, 6958, 5, sim_user_user_optimized)
```

```
# Building the dataframe for above recommendations with columns "song_id" and "predicted_play_count"
pd.DataFrame(recommendations, columns = ['song_id', 'predicted_play_count'])
```

	song_id	predicted_play_count
0	5531	2.553335
1	317	2.518269
2	4954	2.406776
3	8635	2.396606
4	5943	2.390723

User-User Optimized

The tuned User-User collaborative filtering model generated top-5 recommendations with predicted play counts consistently above 2, showing strong similarity-based matches

This highlights how tuning improves the model's ability to recommend songs that align closely with the user's past listening behavior

APPENDIX

```
# Getting top 5 recommendations for user_id 6958 using "svd_optimized" algorithm
svd_recommendations = get_recommendations(df_final, 6958, 5, svd_optimized)
```

```
# Ranking songs based on above recommendations
ranking_songs(svd_recommendations, final_play)
```

	song_id	play_freq	predicted_play_count	corrected_play_count
2	7224	107	2.601899	2.505225
1	5653	108	2.108728	2.012502
4	8324	96	2.014091	1.912029
0	9942	150	1.940115	1.858465
3	6450	102	1.952493	1.853478

Matrix Factorization (SVD) Optimized
The optimized SVD model successfully generated predictions both for songs with known baseline play counts and for songs the user has not interacted with before

The top-5 recommendations show consistent predicted play counts, reflecting the model's ability to identify songs with high potential engagement