ARTIST RECOMMENDER SYSTEM: PERSONALIZING MUSIC DISCOVERY

BY: ADRIAN CASABAL-RAPHAEL, FACUNDO HERRERA RIVAROLA, EHIANE OIGIAGBE, CHIBUIKE UGWU, JUSTIN SIMM, ETHAN VILLALOVOZ

PROBLEM AND MOTIVATION

The Challenge

- Modern streaming platforms offer millions of artists. Users face significant "choice overload" when finding new music.

- Many potential favorites remain undiscovered due to overwhelming options.

Our Solution

- We developed a personalized artist recommender system using collaborative filtering techniques.

- Our approach leverages Last.fm dataset to identify patterns in listening behavior.

DATASET ANALYSIS

Last.fm Dataset from GroupLens

Contains Username, Artist, Track, and Album variables.

Data Characteristics

Sparse matrix with many missing values. Normalized play counts for comparable metrics.

Preprocessing Steps

Filtered low-interaction users and artists. Created user-artist interaction matrix.

Implementation

Python-based data processing pipeline with pandas and numpy libraries.

COLLABORATIVE FILTERING APPROACH

Prediction Engine

Recommends artists based on similar users' preferences

SVD Algorithm

Singular Value Decomposition from Surprise library

Latent Factors

Discovers hidden patterns in user-artist interactions

User-Artist Matrix

Foundation for identifying relationship patterns

MODEL TRAINING PROCESS

Data Preparation

Split data 80/20 for training/testing.

Parameter Tuning

Optimized number of factors, learning rate, and regularization parameters.

Model Training

Used SVD to learn latent factors representing users and artists.

Validation

Evaluated model performance on test set using RMSE metrics.

EVALUATIONS & PREDICTIONS

Quantitative Metrics

Measured prediction accuracy using Root Mean Squared Error (RMSE)

Lower RMSE indicates better prediction quality. Our model achieved a score of 0.0498

```
RMSE: 0.0498
Test RMSE: 0.049809232764141016
Evaluating RMSE of algorithm SVD on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.0530  0.0502  0.0581  0.0486  0.0468  0.0514  0.0040
Fit time          0.11    0.11    0.12    0.11    0.11    0.11    0.01
Test time         0.01    0.04    0.02    0.02    0.01    0.02    0.01
Avg CV RMSE: 0.05135085305117264
Top 5 recommendations for Babs_05: ['Depeche Mode', 'Tim Hecker', 'James Blake', 'Mary Lou Williams', 'Autechre']
```

TOP 5 RECOMMENDATIONS FOR USERS

Top 5 for jonocole:

Rank	Recommended Artist
1	Madlib
2	Sophie
3	Radiohead
4	Dorian Electra
5	Foo Fighters

Top 5 for massdosage:

Rank	Recommended Artist
1	Tim Hecker
2	Dorian Electra
3	Foo Fighters
4	Ocean Waves For Sleep
5	Autechre

Top 5 for Knapster01:

Rank	Recommended Artist
1	Depeche Mode
2	Bicep
3	James Blake
4	The Beatles
5	Foo Fighters

Top 5 for Orlenay:

Rank	Recommended Artist
1	Madlib
2	Depeche Mode
3	Metric
4	Tim Hecker
5	Bill Wurtz

Top 5 for eartle:

Rank	Recommended Artist
1	Ólafur Arnalds
2	Mary Lou Williams
3	Dorian Electra
4	Johann Sebastian Bach
5	ABBA

CHALLENGES

Cold Start Problem

- Recommending artists to new users or recommending new artists remains difficult due to lack of historical data.
- Without listening history, predictions become less accurate.

Solutions

- Filtered out low-interaction users and artists to focus on reliable data.
- Used **matrix factorization (SVD)** to uncover hidden patterns and handle missing values effectively.

Data Sparsity

- Many users listen to only a few artists, leaving much of the interaction matrix empty.
- Sparse data reduces overlap between users, making it harder to find reliable patterns.

KEY FINDINGS AND FUTURE WORK

Key Findings

- SVD-based collaborative filtering produced meaningful, personalized recommendations.
- Switching from numeric user IDs to usernames improved interpretability and user relevance.

Improvements To Make

- Cold-start remains a challenge, new users and artists still need better coverage.
- Plan to integrate Spotify API to leverage artist genres, popularity, and related artists for richer recommendations.
- Explore a hybrid approach combining collaborative and content-based filtering.
- Develop a Streamlit web app to provide an accessible and interactive user interface.

QUESTIONS?