Web Data Integration
Open Music Data Integration

Anh-Nhat Nguyen[2034311], Ching-Yun Cheng[2112322], Shamalan Rajesvaran[2115475], Yen-An Chen[2113612], and Phelan Lee Yeuk Bun[2053019]

Team 1

1 Introduction

In this Web Data Integration project, we aim to consolidate and analyze standardization from multiple sources to gain comprehensive insights into music streaming trends, track performance, and audience preferences. The project leverages three key datasets:

1. Million Song Dataset with Spotify and Last.fm Features Dataset (csv) [4]: This dataset is an enriched version of the Million Song Dataset, a large-scale music database containing detailed metadata and audio features for over 50,000 tracks with 21 attributes. It integrates additional attributes from Spotify and Last.fm, including audio features like danceability, energy, loudness, and popularity metrics such as tags (list attribute), preview URLs, and genre classifications. The merging of these three data sources provides a comprehensive view of each song, making it suitable for analyzing music trends, listener behaviors, and track popularity across platforms.

2. Apple Music Tracks (csv) [2]: This dataset contains detailed information on 10,000 tracks with 24 attributes sourced from Apple Music. It includes attributes such as artist names, album titles, track features (e.g., tempo, key, mode), and genre classifications. In addition to audio metadata, the dataset provides insights into song popularity metrics and trends across the platform. It is ideal for exploring the characteristics of songs on Apple Music, understanding artist performance, and analyzing trends in genres and musical features.

3. Openmusic API Dataset (json) [3]: This dataset offers details about over 5,500 tracks via web APIs (retrieved by /explore & /album?id=<AlbumID>), including track and album metadata, artist information, and playback types (clean/explicit). The use case for this dataset revolves around leveraging real-time API data for in-depth analysis of track consumption patterns, artist popularity, and changes in audience preferences over time.

Together, these datasets will help provide a holistic view of how various musical, commercial, and audience factors contribute to the success of music tracks on different platforms based on various algorithm approach [1].

2 Data Collection and Data Translation

2.1 Data Collection and Dataset

2.1.1 Overview of the Datasets The dataset was obtained from Kaggle in the form of *csv* and OpenMusic API in the form of *csv* and *json*. An overview of dataset attributes is presented in Table 1 below.

Table 1: Dataset structure

Dataset	No Entities	No. Attributes	Attributes
Million Song Dataset with Spotify and Last.fm Features	50,683	21	Track ID, Name, Artist, Spotify Preview URL, Spotify ID, Tags, Genre (*MV 56%*), Year, Duration MS, Danceability, Energy, Key, Loudness, Mode, Speechiness, Acousticness, Instrumentalness, Liveness, Valence, Tempo, Time Signature
Apple Music Tracks	10,000	24	Artist ID, Artist Name, Collection Censored Name, Collection ID, Collection Name, Collection Price, Content Advisory Rating (*MV 85%*), Country, Currency, Disc Count, Disc Number, Is Streamable, Kind, Preview URL, Primary Genre Name, Release Date, Track Censored Name, Track Count, Track Explicitness, Track ID, Track Name, Track Number, Track Price, Track Time (Milliseconds)
Open Music	5,558	18	ShelfTitle, AlbumId, AlbumName, AlbumArtwork, AlbumType, AlbumYear, ArtistId, ArtistName, ArtistProfilePhoto, ArtistSubscribers, TrackId, TrackTitle, TrackPlaybackClean, TrackPlaybackExplicit, TrackLength, TrackIndex, TrackViews, TrackFeatures

2.1.2 Data Preprocessing Data preprocessing steps included filtering irrelevant attributes and normalizing data types.

2.2 Schema Mapping

2.2.1 Design of the Integrated Schema The three datasets contain multiple overlapping attributes as seen in Table 2 below. There are 5 attributes within our integrated schema that overlap across at least 2 of 3 input schemata, namely "Artist", "Track Name", "Genre", "Track Duration" and "Release Date".

Table 2: Table of Integrated Schema Attributes

Attribute Name	Datatype	Datasets in which attribute found
Track	string	Spotify Musicality, Spotify Streaming Statistics, Openmusic
Artist	string	Spotify Musicality, Spotify Streaming Statistics, Openmusic
Album	string	Spotify Musicality, Spotify Streaming Statistics, Openmusic
Youtube Views	decimal	Spotify Streaming Statistics, openmusic
YouTube Likes	decimal	Spotify Streaming Statistics
Release Date	datetime	Spotify Streaming Statistics, openmusic
Danceability	decimal	Spotify Streaming Statistics
Energy	decimal	Spotify Streaming Statistics
Key	decimal	Spotify Streaming Statistics
Loudness	decimal	Spotify Streaming Statistics
Speechiness	decimal	Spotify Streaming Statistics
Acousticness	decimal	Spotify Streaming Statistics
Instrumentalness	decimal	Spotify Streaming Statistics
Liveness	decimal	Spotify Streaming Statistics
Valence	decimal	Spotify Streaming Statistics
Tempo	decimal	Spotify Streaming Statistics
TrackPlaybackClean	string	openmusic
TrackPlaybackExplicit	string	openmusic

2.2.2 Tools and Challenges Tools used include Altova MapForce. Challenges encountered during schema mapping included aligning attributes and handling missing data.

2.2.3 Conversion to Target Schema Datasets were converted to the target schema resulting in XML files.

3 Phase II: Identity Resolution

3.1 Initiate Gold Standard

3.1.1 Method for Building the Gold Standard The gold standard was built using a sampling method ensuring matches, non-matches, and corner cases.

3.2 Challenges with Gold Standard and Improvement

3.2.1 Edit-Distance with Single Key Edit-distance with a single key (Track) did not have high coverage. Solutions included using multiple keys and advanced matching techniques.

3.2.2 Selection Bias Addressed selection bias by ensuring diverse and representative samples.

3.2.3 Performance Improvement Implemented blocking, Bloom Filtering, and used LLM for performance improvement.

3.3 Matching Strategies

3.3.1 Blocking Methods To reduce unnecessary comparisons during identity resolution (IR), we generated blocking keys based on track names. The blocking key was derived from the bigrams of the first three tokens of each track name. We then experimented with two blocking methods: **Standard Blocking** and the **Sorted Neighborhood Method**, both using these track name-based blocking keys.

In the case of running IR for the apple + opendb datasets, the maximum number of entities sharing the same hashed blocking key was 9.696 (for the blocking key "CH"), while the minimum was 1 (for the blocking key "LAINTH"). As for the million + opendb datasets, the maximum number of entities sharing the same hashed blocking key was 67.404 (for the blocking key "RE"), while the minimum was 1.560 (for the blocking key "BL"). When using the Sorted Neighborhood Method, the window size would need to be greater than 9.696 and 67.404 respectively to ensure that no matches were missed. However, such a large window size would significantly increase resource consumption by comparing many irrelevant records.

Given this inefficiency, for both of our entity matching comparisons, we ultimately decided to adopt **Standard Blocking**, which efficiently grouped entities based on their blocking keys without requiring extensive resource expenditure or risking lost matches.

3.3.2 Similarity Metrics As mentioned above, in the entity matching for apple + opendb, we use **track name, artist name, album name,** and **album year** as the basis for determining entity matching. As for entity matching for million + opendb, since there is no album name attribute in million dataset, we only use **track name, artist name**, and **album year** as the basis. For each attribute with a data type of string, we tested various metrics, including edit-based (Levenshtein, Jaro, Jaro-Winkler), token-based (Jaccard), and phonetic (Soundex). For numeric attribute, we use absolute difference of 2 years.

3.4 Evaluation

3.4.1 Metrics and Analysis After testing over 50+ combinations of comparators for entity matching between apple + opendb datasets and million + opendb datasets, the metrics shown in Table 3 were found to be the most suitable for their respective attributes.

Table 3: Comparators Used for Entity Matching Across Datasets

Dataset	**Attribute (Comparator)**	**Reason for Effectiveness**
apple+opendb; million+opendb	Track Name (Jaccard)	Track names often include variations like additional descriptors, e.g., single, feat.
apple+opendb	Artist Name (Jaro-Winkler)	Artist names are short and structured. Most typos occur in the last name rather than the first name.
million+opendb	Artist Name (Equal)	Many entities in million + opendb have the same track name but different artist names. Within those entities, the artist names are quite similar in some cases; therefore, the equal comparator is needed to clearly distinguish ambiguous entities.
apple+opendb	Album Name (Jaccard)	Most of the singles use the track name as their album name, so the same pattern applies here.
apple+opendb; million+opendb	Album Year (Absolute Difference ± 2)	Accounts for real-world scenarios like re-releases and recording/release year discrepancies.

We organized some of our IR tests in Table 4. Those combination we finally chose are highlighted in yellow. In the results for apple + opendb, we intuitively selected the one that demonstrated the best performance across Precision, Recall, and F1-score. Upon closer inspection of the correspondences, the matches were indeed highly accurate, confirming our choice.

However, in the results for million + opendb, after reviewing the correspondences, we decided to select the option that did not achieve the best performance metrics. This decision was based on the presence of more ambiguous data in this comparison, which, combined with an insufficient golden standard to accurately use the album year to distinguish different entities, resulted in false positives within the correspondences. Under these circumstances, the selected option demonstrated the most balanced precision and recall, minimizing false positives while still capturing the majority of true matches. This balance made it the optimal choice for this dataset.

Table 4: Entity Matching Benchmark Table

Datasets	**Matching Rule**	**B**	**P**	**R**	**F1**	**# Corr**
apple+opendb	Track (J*), Artist (JW*), Album (J), Album Year (2Y*)	SNB Track (20)	0,99	0,66	0,79	85
apple+opendb	Track (J), Artist (JW), Album (J), Album Year (2Y)	SNB Track (60)	0,99	0,82	0,89	105
apple+opendb	Track (J), Artist (JW), Album (J), Album Year (2Y)	Standard Track	0,99	0,93	0,96	120
apple+opendb	Track (L), Artist (JW), Album (L*), Album Year (2Y)	Standard Track	0,99	0,92	0,9544	118
apple+opendb	Track (S*), Artist (JW), Album (S), Album Year (2Y)	Standard Track	0,97	0,95	0,96	170
apple+opendb	Track (J), Artist (J), Album (J)	Standard Track	0,87	0,94	0,91	291
million+opendb	Track (J), Artist (Equal), Album Year (2Y)	SNB Track (20)	0,94	0,72	0,82	684
million+opendb	Track (J), Artist (Equal), Album Year (2Y)	SNB Track (60)	0,92	0,85	0,88	835
million+opendb	Track (J), Artist (Equal), Album Year (2Y)	Standard Track	0,92	0,94	0,93	942
million+opendb	Track (J), Artist (JW), Album Year (2Y)	Standard Track	0,91	0,99	0,95	2.078
million+opendb	Track (S), Artist (JW), Album Year (2Y)	Standard Track	0,88	0,94	0,91	1.326
million+opendb	Track (J), Artist (Equal)	Standard Track	0,88	0,94	0,91	1.082

***Note:** J: Jaccard; JW: Jaro-Winkler; 2Y: Absolute Difference ±2; S: Soundex; L: Levenshtein

Although the data in the table suggests that the results of Entity Matching are reasonably satisfactory, a closer examination of the million + opendb correspondences reveals that the differentiation of entities based on the album year is insufficient. This limitation is also reflected in the results of our Group Size analysis.

Table 5: Group Size Analysis

Group Size	**2**	**3**	**4-7**	**8-13**	**14**
Frequency	650	104	45	4	0
Distribution	81%	13%	6%	0%	0%

3.4.2 Error That Remain Our IR results for album years continue to show inconsistencies, with mismatched or slightly inaccurate years persisting even when using a tolerance of ± 2 years. These discrepancies are particularly evident when dealing with re-releases, remasters, or cases where album metadata varies significantly across datasets.

We think that the root cause of this issue lies in the golden standard we employed, which was derived by concatenating track name, artist name, and album year into a single string and performing an initial comparison using the edit-based Levenshtein metric. Since album years were not accurately distinguished in this process, it led to the observed inconsistencies. This misalignment in album years will negatively impact attribute consistency during the Data Fusion phase. If this issue remains unresolved, the resulting fused data will likely inherit these inconsistencies, compromising its overall quality and trustworthiness.

4 Data Fusion

4.1 Fusion Rules

4.1.1 Conflict Resolution Strategies In the data fusion process, conflict resolution strategies are essential to derive the most accurate and reliable representation of records when combining multiple datasets. Our conflict resolution strategies include using reliable datasets with sufficient attribute intersection for fusion, as well as strong identity resolution results that maintain quality of data. We decided to use all of our datasets (apple, million, opendb) as they fulfil our prerequisites with IR results as shown above. We had assigned a provenance score to each dataset in order to reflect its reliability and accuracy. 'apple' dataset was given the highest score (3.0), followed by 'million' (2.0), and then 'opendb' (1.0). This scoring system allowed the fusion process to favor attributes from the more reliable datasets when resolving conflicts.

4.1.2 Specific Fusion Rules Our team have included a specific fusion rules for resolving conflicts in key attributes to ensure that the fused dataset accurately reflects the most reliable, comprehensive, and meaningful information. These rules are tailored to the nature of each attribute and the type of conflicts typically encountered. For the track names, album names and artist names, the longest string strategy is used to resolve conflicts. This approach ensures that the most descriptive and complete track title is chosen, as longer names often include additional context, such as versions or remixes. Likewise, this fusion rule will retain information such as featuring artists or collaborations for the artist names.

For the album year, a voting strategy is employed for resolving conflicts. This strategy aggregates values across the 3 datasets and selects the most commonly occurring year, reflecting a consensus among the sources. This minimizes the impact of outliers or errors in individual datasets and ensures consistency.

The favor source with the highest provenance score strategy is used for the track duration, avoiding discrepancies caused by slight rounding differences or errors in less reliable sources.

4.2 Fused Data Output

4.2.1 Post-Fusion Dataset Post-fusion dataset size and density improvements were noted. Examples of fused records were provided.

4.3 Quality Evaluation

We conducted extensive testing with various combinations of fusion methods. Table 6 shows four representative strategies that combine different fusion methods, including LongestString(), FavourSource(), ShortestString(), and MostRecent(). Strategy 4 demonstrated the highest accuracy at 84% and was therefore selected as our optimized fusion strategy.

To evaluate our optimized fusion strategy's effectiveness, we analyzed its performance across different attributes (Table 7). While most attributes achieved high accuracy, Album Name information showed notably low accuracy (50%) due to the frequent occurrence of the same track appearing in different album releases (e.g., singles, original albums, compilation albums). This common practice in the music industry creates challenges in determining the canonical album name for a track, leading lower accuracy and consistency scores for Album Name (50%, 81%) and AlbumYear (75%, 56%) attributes.

5 Conclusion and Future Work

5.1 Limitations of the Project

Discussed limitations such as incomplete data and computational constraints.

5.2 Recommendations for Future Improvements

Recommendations included using advanced ML models for identity resolution.

Table 6: Data Fusion Strategy Comparison

Fusion Strategy	Method	Accuracy
Strategy 1	ArtistFuserLongestString(), AlbumYearFuserVoting(), AlbumFuserLongestString(), DurationFuserFavourSource(), TitleFuserLongestString(), ExplicitnessFuserFavourSource()	80%
Strategy 2	ArtistFuserFavourSource(), AlbumYearFuserFavourSource(), AlbumFuserFavourSource(), DurationFuserFavourSource(), TitleFuserFavourSource(), ExplicitnessFuserFavourSource()	80%
Strategy 3	ArtistFuserShortestString(), AlbumYearFuserVoting(), AlbumFuserShortestString(), DurationFuserFavourSource(), TitleFuserShortestString(), ExplicitnessFuserFavourSource()	82%
Strategy 4 (Optimized)	ArtistFuserShortestString(), AlbumYearFuserMostRecent(), AlbumFuserShortestString(), DurationFuserMostRecent(), TitleFuserShortestString(), ExplicitnessFuserMostRecent()	84%

Table 7: Optimized Fusion Strategy Performance

Attribute	Method	Accuracy	Consistency	Density
Artist	ArtistFuserShortestString()	95%	100%	100%
AlbumYear	AlbumYearFuserMostRecent()	75%	56%	100%
Album	AlbumFuserShortestString()	50%	81%	100%
Duration	DurationFuserMostRecent()	85%	88%	100%
Track	TitleFuserShortestString()	85%	98%	100%
TrackExplicitness	ExplicitnessFuserMostRecent()	90%	99%	100%

References

1. AnHai Doan, Alon Halevy, and Zachary Ives. *Principles of Data Integration*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, June 2012.

2. Kanchana1990. Song dataset: 10,000 apple music tracks. URL: https://www.kaggle.com/datasets/kanchana1990/apple-music-dataset-10000-tracks-uncovered.
3. OatsCG. OatsCG/Openmusic-Server-Specs, September 2024. original-date: 2024-01-10T01:36:38Z. URL: https://github.com/OatsCG/Openmusic-Server-Specs.
4. UndefineNull. Million song dataset + spotify + last.fm. URL: https://www.kaggle.com/datasets/undefinenull/million-song-dataset-spotify-lastfm.