Faculty Computer Science

Data Science

Moscow 2024

Building a Music Recommendation System

WSDM - KKBox's Music Recommendation Challenge

Presented by Shkliar Mikhail
MNOD231
23.12.24

Domain Description

Overview:

- Music streaming is a rapidly growing industry.
- Personalization is key to user retention and satisfaction.

Challenges:

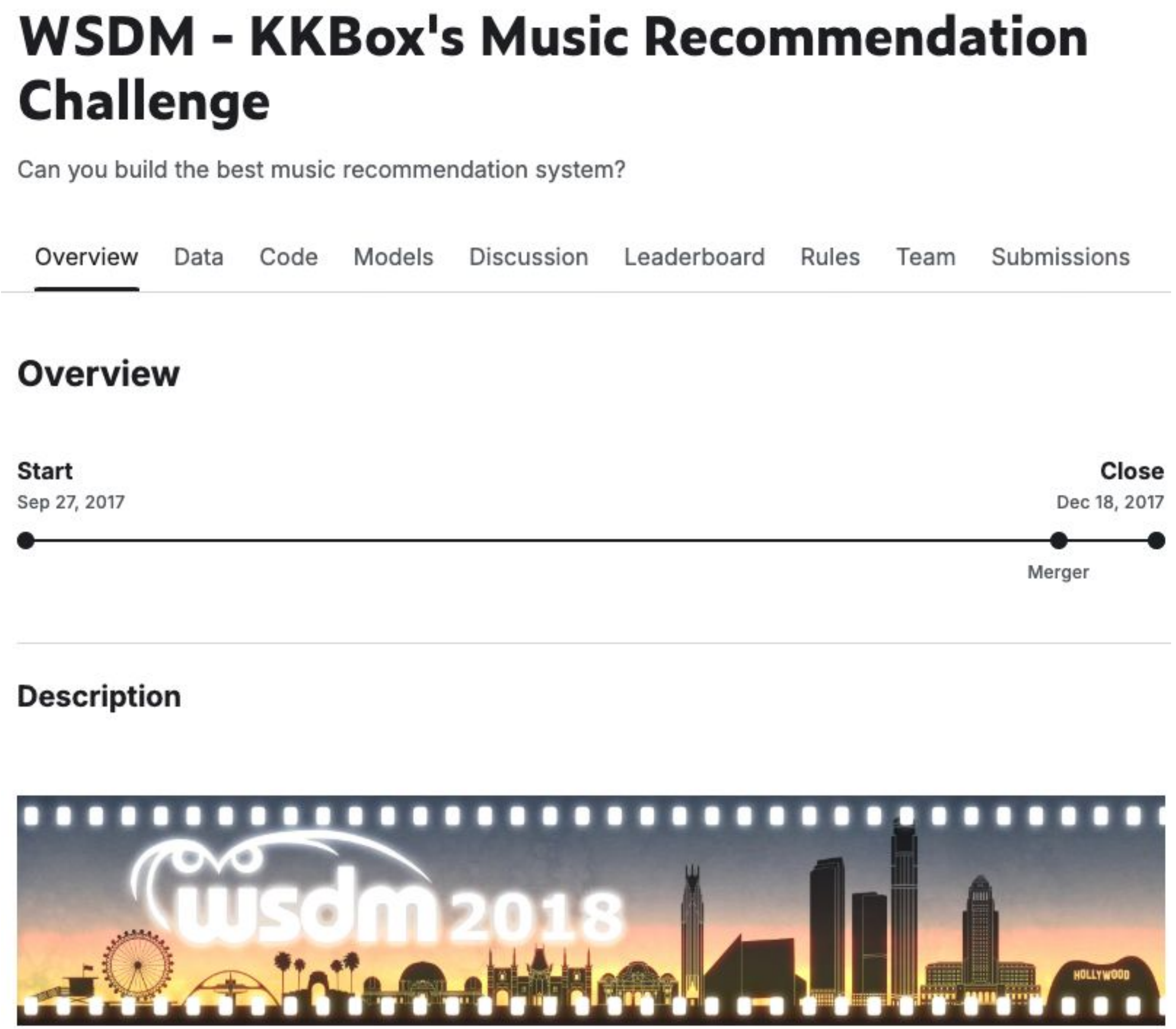
- Predicting user preferences for new songs and new users (cold start).
- Handling large-scale and complex datasets.

KKBox:

- Asia's leading streaming platform.
- Over 30 million tracks in their library.

Goals and Objectives

- **Main Goal:** Predict if a user will replay a song within a month.
- **Key Objectives:**
  - Understand and preprocess user, song, and event metadata.
  - Engineer and select features to optimize model performance.
  - Train and evaluate various machine learning models.
  - Provide actionable insights to improve KKBox’s recommendations.



Relevance of the Work

- Why This Matters:
  - Personalized recommendations improve user experience and retention.
  - Addressing cold start problems benefits platforms and users.
  - Enhances user engagement with diverse musical content.
- Broader Impact:
  - Machine learning advancements for large-scale recommendation systems.

Dataset Overview

- **Key Components:**
  - User Data: demographics, registration method, activity.
  - Song Data: length, genre, artist, language.
  - Event Data: playback source, tab, and screen type.

- **Target Variable:**
  - 1: User replayed the song within a month.
  - 0: User did not replay the song.

- **Visual:** Diagram summarizing data relationships (e.g., user → listens to → song).

kkbox-music-recommendation-challenge	--
test.csv	347,8 МБ
songs.csv	221,8 МБ
song_extra_info.csv	181 МБ
sample_submission.csv	29,6 МБ
members.csv	2,5 МБ
train.csv	971,7 МБ

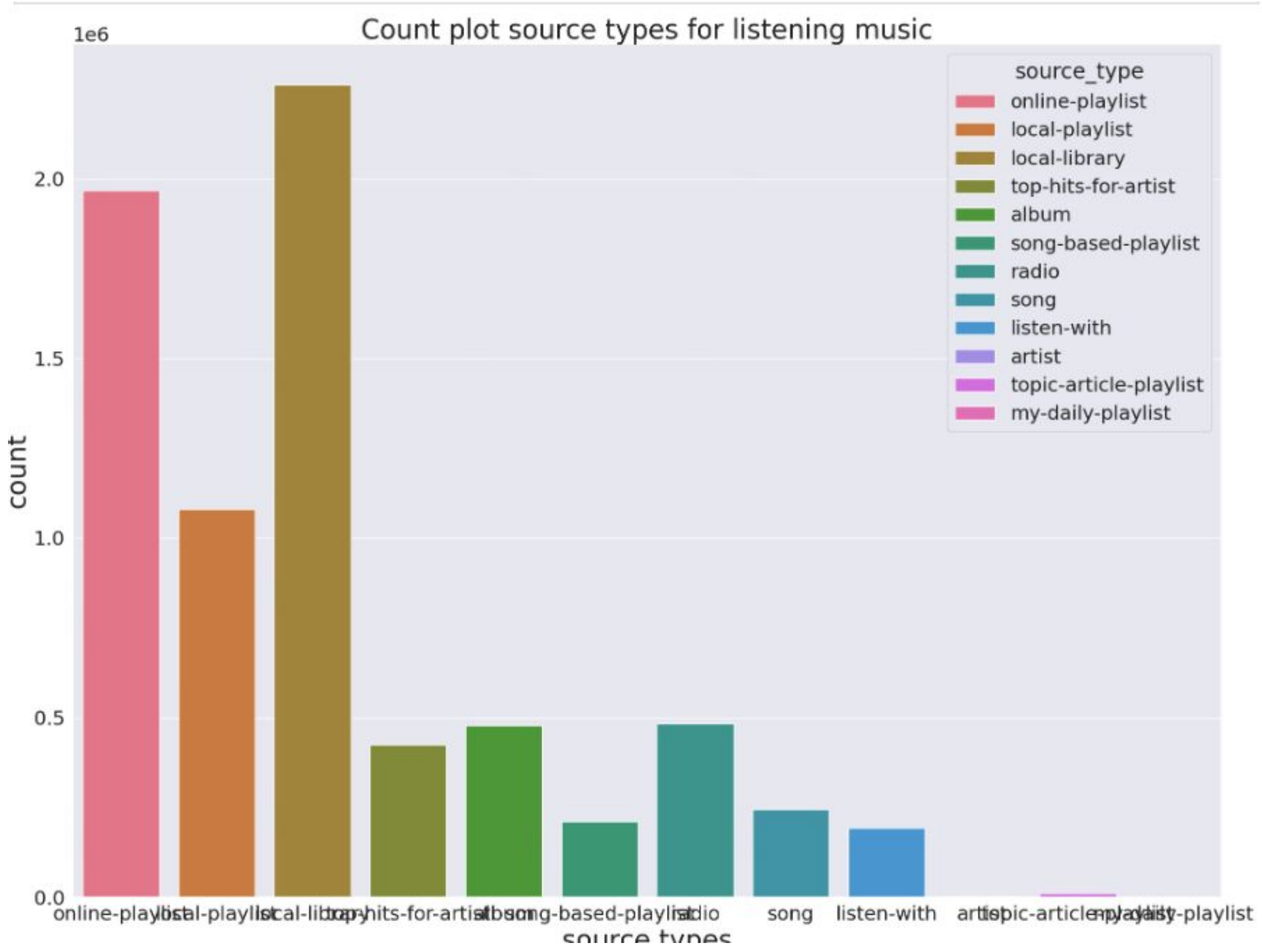
Publications analysis

Publication	Main ideas	Restrictments	Conclusions
"Deep Content-based Music Recommendation"	The use of Convolutional Neural Networks (CNN) for analyzing spectrograms of music tracks results in high prediction accuracy based on the content of the tracks.	It does not take into account social and contextual aspects of music perception, such as track popularity or seasonal listener preferences.	Combining content-based analysis with other approaches will improve the accuracy of recommendations.
"Collaborative Filtering for Music Recommendation"	The application of collaborative filtering methods to uncover hidden patterns in music preferences based on user ratings.	Issues of scalability, cold start, and data sparsity.	Hybrid systems that combine collaborative filtering with content-based approaches improve the quality of recommendations.

Data Preprocessing - Overview

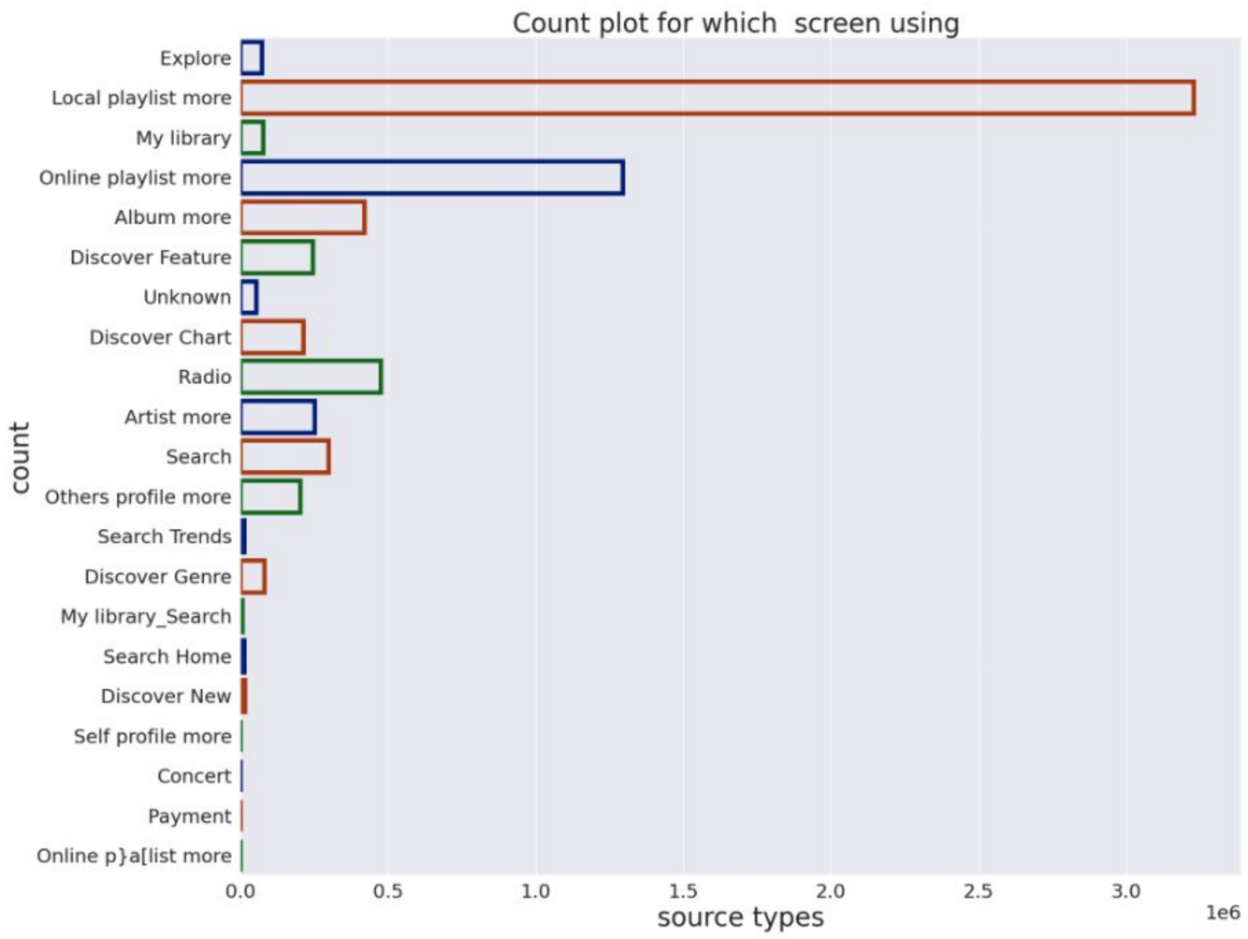
- **Steps Taken:**
  - Handling missing values.
  - Encoding categorical variables.
  - Merging datasets (e.g., user, song, and event data).
  - Dealing with outliers in features like bd (age).
  - Standardizing numerical features.

- **Challenges:**
  - Imbalanced target variable.
  - High cardinality in categorical data.



Exploratory Data Analysis - Source Types

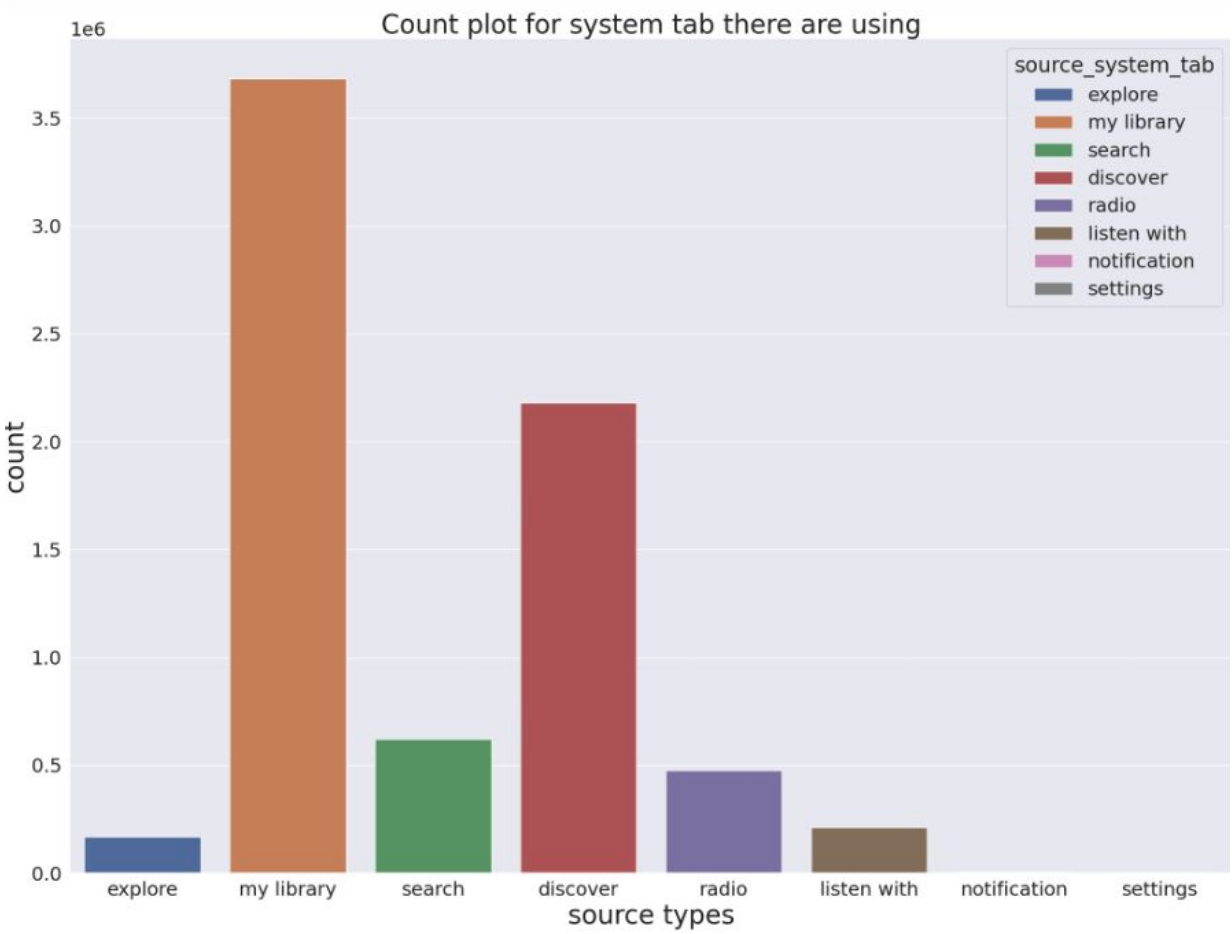
- **Insights from Source Types:**
  - Local library and online playlist are the most common ways users interact with music.
  - Other categories like "album" and "song-based playlist" have less interaction.
- **Purpose:**
  - Understand user behavior.
  - Guide feature engineering for models.



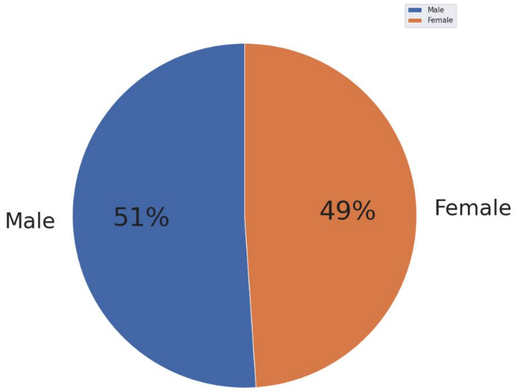
Exploratory Data Analysis - System Tabs

- **Insights from System Tabs:**
  - "My Library" is the most accessed tab, indicating strong user loyalty to saved content.
  - Tabs like "Discover" and "Search" are used less frequently, which might reflect limited exploration behavior.

- **Impact:**
  - Helps identify key features related to user preferences.

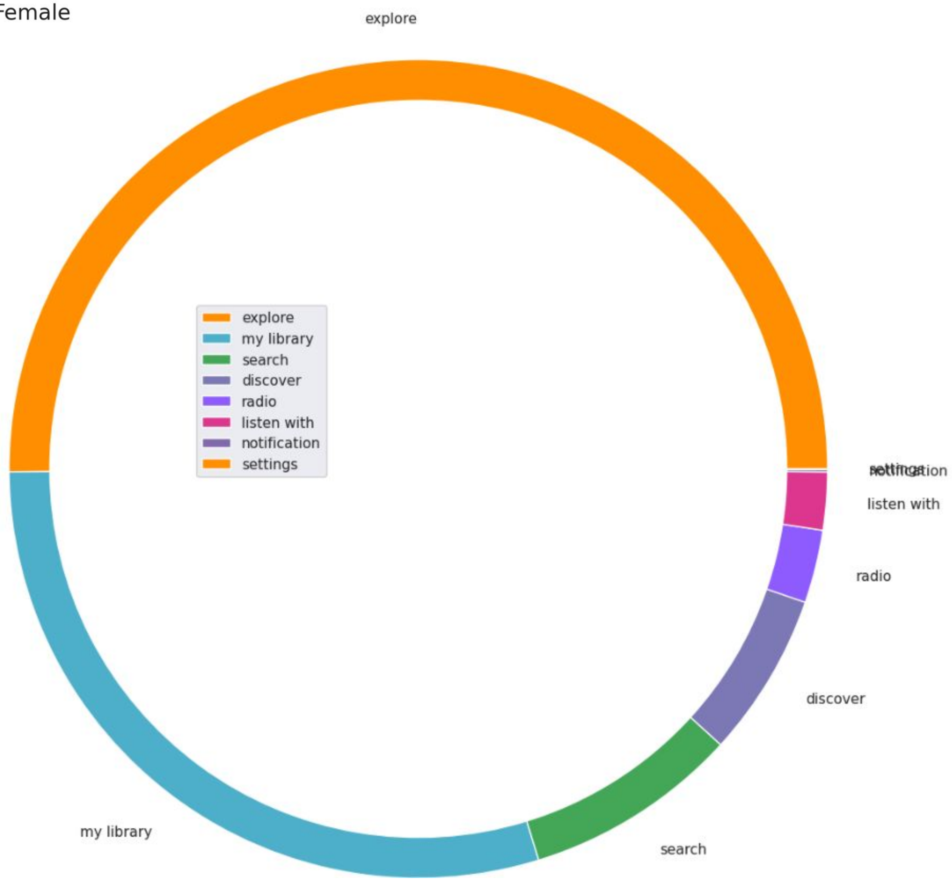


Gender Distribution

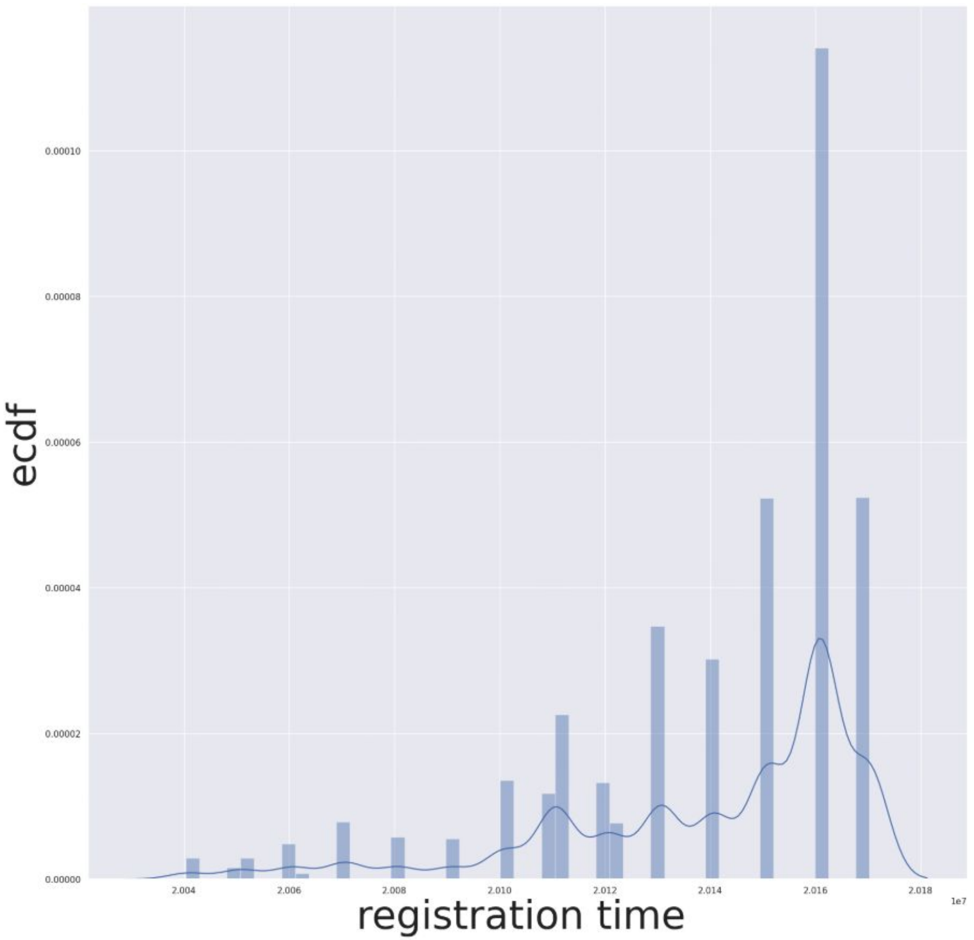


- **Distribution:**
  - Male users constitute 51%.
  - Female users account for 49%.

- **Purpose:**
  - Understand user demographics.
  - Explore potential gender-based differences in behavior.



Registration Time Analysis



Insights:

- Most users registered between 2012 and 2016.
- Registration times are right-skewed, suggesting a growing user base over time.

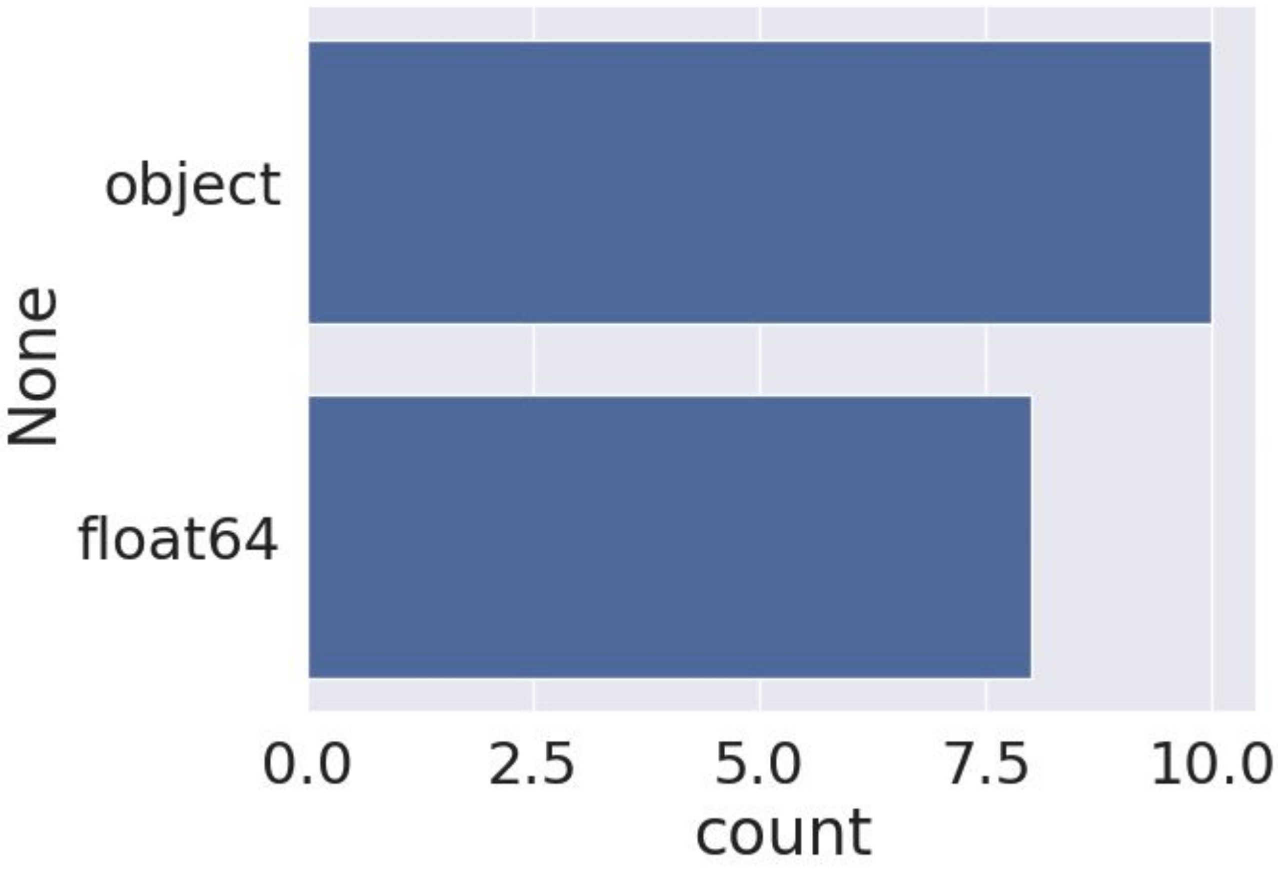
Purpose:

- Highlight user growth trends.
- Inform potential time-based features.

Source and Screen Usage

- **Insights:**
  - Local playlists and "My Library" dominate both screen and source usage.
  - Other features, such as "Discover" and "Online Playlist," have significant but lesser usage.

- **Purpose:**
  - Understand user interaction patterns.
  - Prioritize features for recommendation improvement.



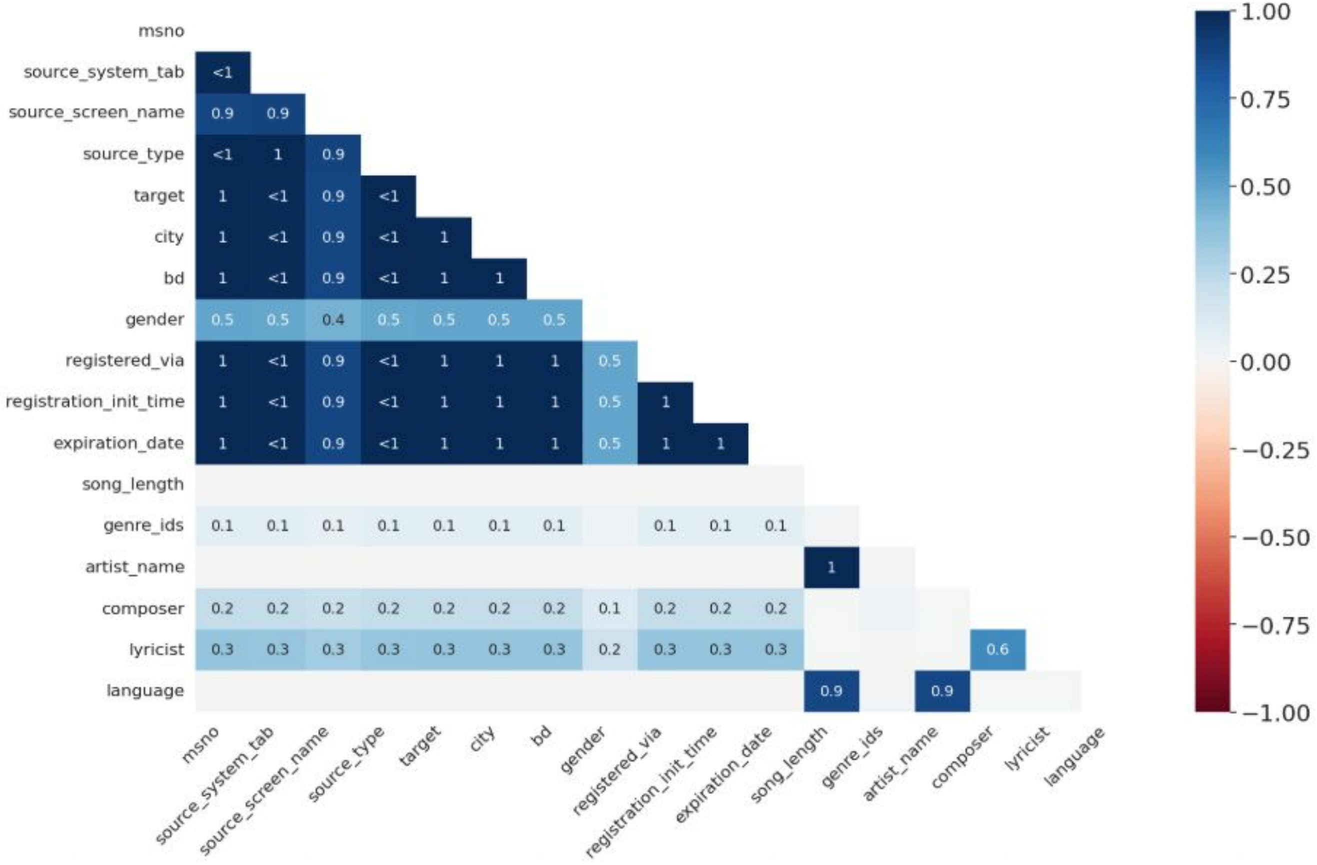
Analysis of Missing Values - Heatmap

Key Observations:

- Missingness is concentrated in certain features like gender, composer, and lyricist.
- Strong correlation between missing gender and other features like city and target.

Perpose:

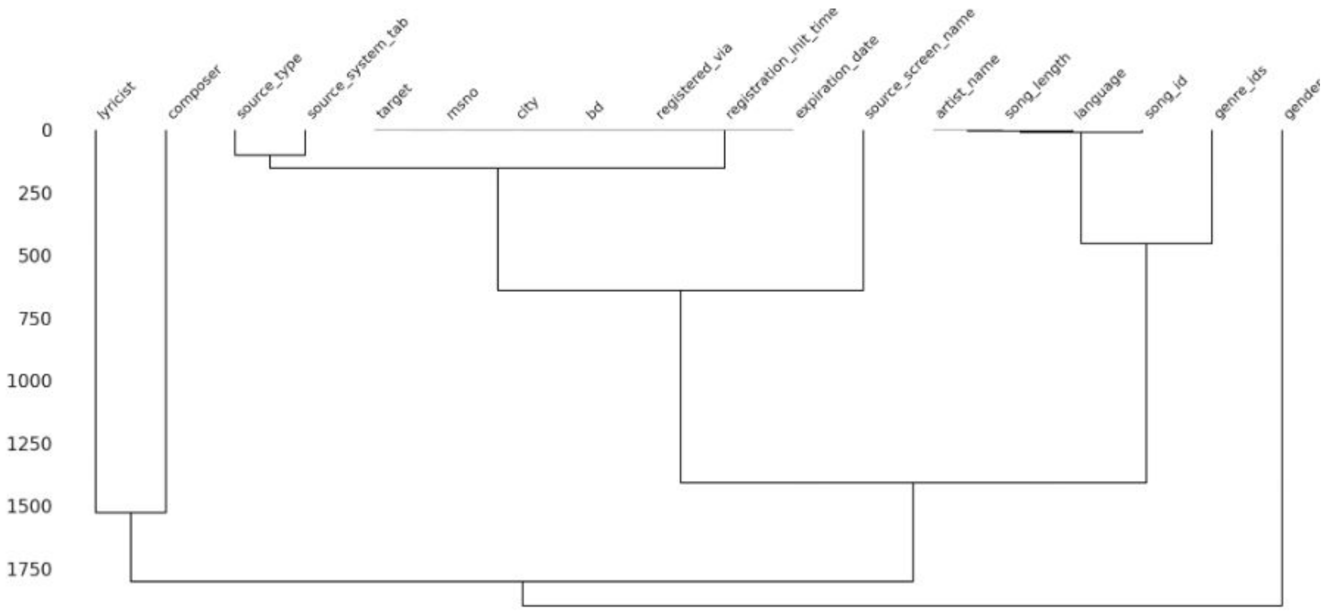
- Identify patterns in missing values.
- Guide imputation strategies.



Analysis of Missing Values - Dendrogram

Key Observations:

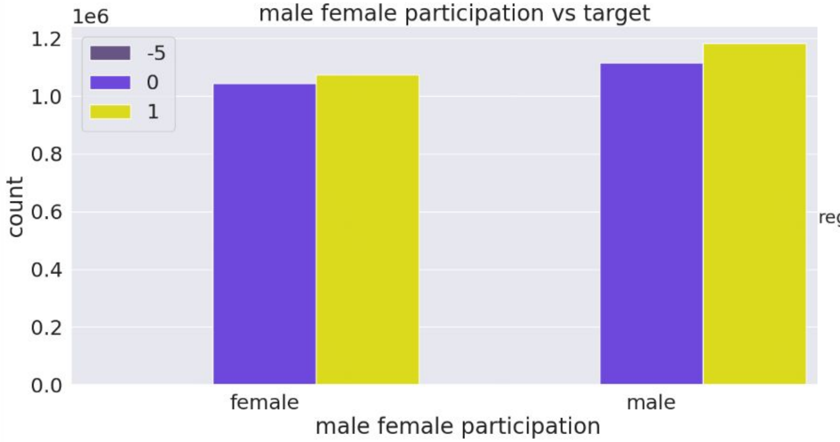
- Hierarchical clustering reveals relationships between missing patterns.
- Features like song_id, language, and song_length are closely related in terms of missing data.

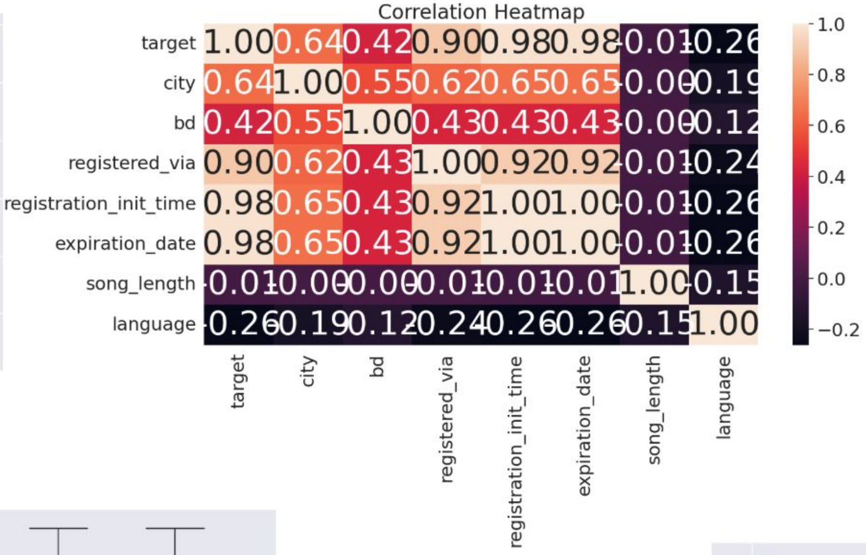


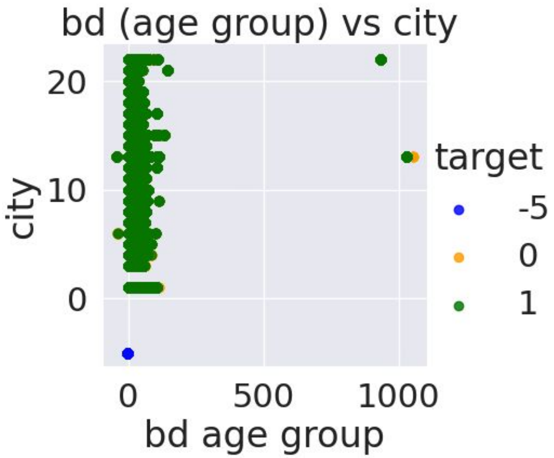
Impact:

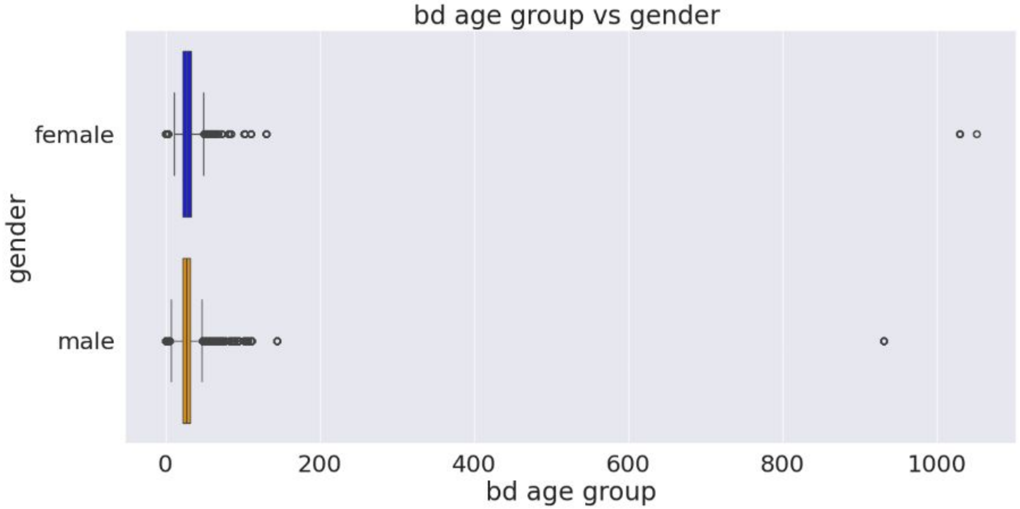
- Helps prioritize features for imputation or exclusion.
- Identifies clusters for group-wise handling.

Demographic and Behavioral Analysis









Feature Selection - Overview

- **Key Features Engineered:**
  - **User Information:**
    - membership_days: Categorized the membership duration of users.
    - registration_year and expiration_year: Extracted from dates for temporal trends.
  - **Song Metadata:**
    - song_year: Categorized song release years derived from ISRC codes.
    - genre_ids: Simplified by selecting the first genre for songs with multiple genres.
  - **Behavioral Data:**
    - repeat_count, play_count, and repeat_percentage: Capture the popularity and re-listen behavior of songs.
    - Similar metrics for artists: repeat_percentage_artist and play_count_artist.
- **Purpose:**
  - Add meaningful features to capture user, song, and artist behavior.
  - Simplify and categorize data for better modeling performance.

```
RangeIndex: 7377418 entries, 0 to 7377417
Data columns (total 20 columns):
 #   Column                   Dtype
---  ------                   -----
 0   msno                     object
 1   song_id                  object
 2   source_system_tab        category
 3   source_screen_name       category
 4   source_type              category
 5   target                   uint8
 6   artist_name              category
 7   genre_ids                object
 8   language                 category
 9   city                     category
 10  registered_via           category
 11  registration_year        int64
 12  expiration_year          int64
 13  membership_days          category
 14  song_year                float64
 15  repeat_count             int64
 16  play_count               int64
 17  repeat_percentage        float64
 18  play_count_artist        float64
 19  repeat_percentage_artist float64
dtypes: category(8), float64(4), int64(4), object(3), uint8(1)
memory usage: 713.3+ MB
```

Feature Selection - Finalization

- **Final Features Used:**
  - User Information: city, registered_via, membership_days, registration_year, expiration_year.
  - Song Information: song_id, genre_ids, language, song_year.
  - Behavioral Metrics: play_count, play_count_artist.
- **Dropped Features:**
  - Redundant intermediate metrics: repeat_count, repeat_percentage, and related artist/user metrics.
  - Unnecessary columns: artist_name, song_length, and detailed ISRC-related fields.
- **Data Cleaning:**
  - Handled missing values for counts and metrics.
  - Ensured consistent data types for categorical and numerical features.

Model Development - LightGBM with Cross-Validation

Why LightGBM?

- Efficient for large datasets with complex features.
- Handles categorical data natively.
- Higher AUC and stability compared to simpler models.
- Flexible hyperparameters and GPU support.

Training Setup:

- 3-fold cross-validation, AUC as metric.
- Key hyperparameters:
- Learning rate: 0.2
- Leaves: 256
- Bagging fraction: 0.95

Results:

- AUC Scores: 0.7706, 0.7284, 0.6878.
- Average AUC: 0.729.
- Total Time: 6122.68 sec (on CPU).

```
Start of training...

Training on split 1 of 3...
[LightGBM] [Warning] Categorical features with more bins than the configured maximum bin number found.
[LightGBM] [Warning] For categorical features, max_bin and max_bin_by_feature may be ignored with a large number o
f categories.
Split 1 is over. Execution time: 798.49 seconds. AUC: 0.7706

Training on split 2 of 3...
[LightGBM] [Warning] Categorical features with more bins than the configured maximum bin number found.
[LightGBM] [Warning] For categorical features, max_bin and max_bin_by_feature may be ignored with a large number o
f categories.
Split 2 is over. Execution time: 736.07 seconds. AUC: 0.7284

Training on split 3 of 3...
[LightGBM] [Warning] Categorical features with more bins than the configured maximum bin number found.
[LightGBM] [Warning] For categorical features, max_bin and max_bin_by_feature may be ignored with a large number o
f categories.
Split 3 is over. Execution time: 750.20 seconds. AUC: 0.6878

Training is over. Overall execution time: 6122.68 seconds.
```

Model Selection & Quality Assessment

Model Evaluation:

- Metric: AUC (Area Under the Curve) for LightGBM.
- Accuracy and AUC assessed for other models using a 10% sample.

Results Summary:

Model	Mean Accuracy	Std Accuracy	Mean AUC	Std AUC
Logistic Regression	0.5719	0.0013	NaN	NaN
Random Forest	0.6693	0.0009	NaN	NaN
XGBoost	0.6554	0.0011	0.7072	0.0015
LightGBM	**0.6810**	**0.0020**	**0.7289**	**0.0338**

Insights:

- **LightGBM** achieves the highest AUC (0.7289), confirming its effectiveness for this dataset.
- **Random Forest** performs well in accuracy but lacks AUC evaluation.
- **XGBoost** shows a balance between accuracy and AUC but underperforms compared to LightGBM.
- **Logistic Regression** struggles with low accuracy and no AUC results.

Final Submission:

- LightGBM predictions were saved to submission.csv for the competition.

Competition Results and Insights

- **Our results:**
    - Private Score: 0.65886
    - Public Score: 0.65642

- **Top Leaderboard Scores:**
    - 1st: 0.74787
    - 2nd: 0.74693
    - 3rd: 0.74688

- **Key Insights:**
    - Leaders used up to 300GB memory, which we couldn't match on a local machine.
    - Memory is critical for detailed feature engineering and model tuning.
    - On train data (fold 1), our AUC was better than the leaders.

WSDM - KKBox's Music Recommendation Challenge

Can you build the best music recommendation system?

Overview Data Code Models Discussion Leaderboard Rules Team Submissions

Submissions

0/2

You selected 0 of 2 submissions to be evaluated for your final leaderboard score. Since you selected less than 2 submissions, Kaggle auto-selected up to 2 submissions from among your public best-scoring unselected submissions for evaluation. The evaluated submission with the best Private Score is used for your final score.

Submissions evaluated for final score

All Successful Selected Errors

Recent

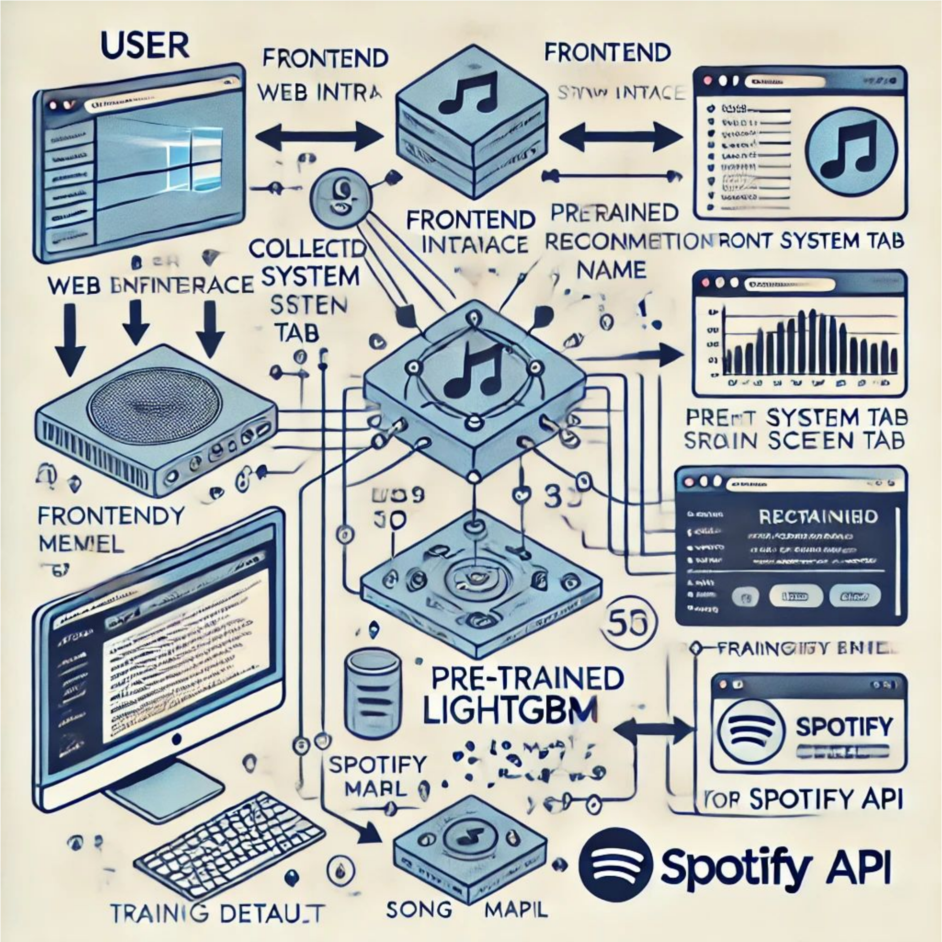
Submission and Description	Private Score	Public Score	Selected
submission.csv Complete (after deadline) · 14h ago · LightGBM	0.65886	0.65642	☐
submission (1).csv Complete (after deadline) · 19h ago · LightGBM model results	0.65854	0.65638	☐

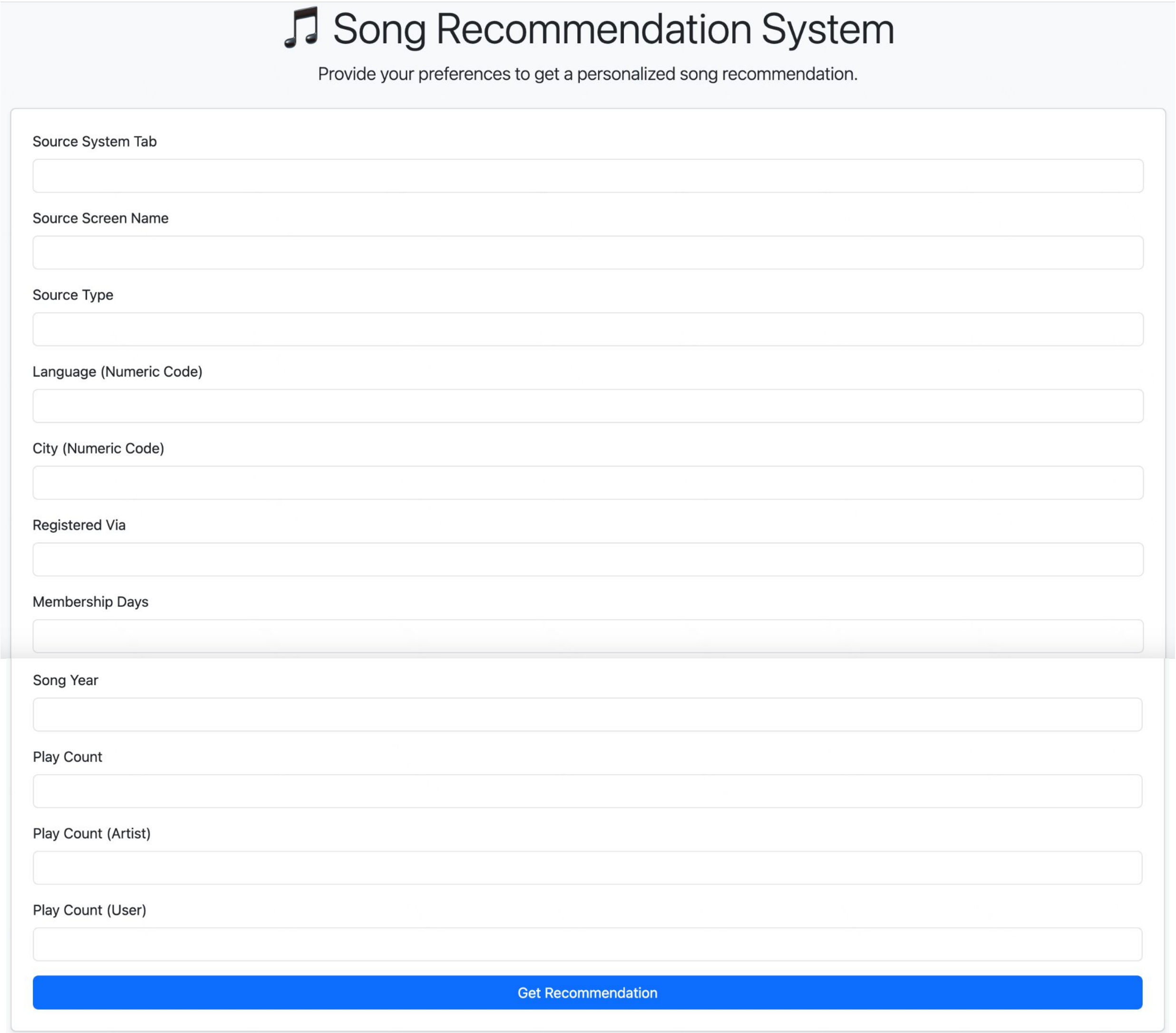
Web Application for Song Recommendation

- **How it works:**
  - User provides their preferences through an input form.
  - The application processes the input and matches it against the training dataset.
  - The pre-trained LightGBM model predicts the best-matching song based on user data.
  - The application displays the song, with a link to stream it directly from Spotify.

- **Key Features:**
  - Simple and user-friendly interface.
  - Predicts personalized recommendations in real-time.
  - Integrates with Spotify to enable direct song playback.

- **Architecture Overview:**
  - **Frontend**: Collects user input via a form and displays the recommended song.
  - **Backend**:
    - Processes the input.
    - Utilizes the LightGBM model to make predictions.
    - Fetches song details from the training dataset and Spotify API.
  - **Model**: Pre-trained LightGBM model for fast and accurate predictions.





Demonstration of web application work

Conclusion

- **Key Achievements:**
  - Developed a personalized song recommendation system using LightGBM.
  - Achieved a competitive AUC of 0.7289 despite resource limitations.
  - Built a scalable and user-friendly web application integrated with Spotify.
- **Key Insights:**
  - Memory and computational resources significantly impact model performance and feature engineering.
  - Advanced machine learning models like LightGBM provide robust solutions for complex datasets.
- **Future Directions:**
  - Optimize the model with larger datasets and enhanced computational resources.
  - Integrate user history and preferences for more personalized recommendations.
  - Explore deep learning methods for further performance improvements.

Sources

1. Kaggle competition: https://www.kaggle.com/competitions/kkbox-music-recommendation-challenge/
2. Tzanetakis, George, et al.: "Music Genre Classification with Machine Learning." Journal of Machine Learning Research - http://jmlr.org/papers/volume13/tzanetakis12a/tzanetakis12a.pdf (Дата обращения: 25.08.2023)
3. Choi, Keunwoo, et al.: "Automatic Tagging using Deep Convolutional Neural Networks." ISMIR - https://ismir.net/archives/2016/Choi_Automatic_Tagging_using.pdf (Дата обращения: 01.09.2023)
4. Spotify Technology S.A.: "Developing Spotify's Music Recommendation Engine." - https://www.spotify.com/us/about-us/contact/ (Дата обращения: 09.09.2023)
5. Baccigalupo, Claudio, and Juan Manuel Pacheco: "Music Recommendation: A Multi-level Perceptual Approach." Artificial Intelligence Review - https://link.springer.com/article/10.1007/s10462-008-9101-4 (Дата обращения: 21.10.2023)
6. Statista: "Global Music Streaming Market Trends and Forecasts (2024–2029)." - https://www.statista.com/statistics/652140/global-music-streaming-revenue/ (Дата обращения: 04.01.2024)
7. Dieleman, Sander, et al.: "End-to-end Learning for Music Audio." IEEE International Conference on Acoustics, Speech, and Signal Processing - https://ieeexplore.ieee.org/document/7952134 (Дата обращения: 12.02.2024)
8. McFee, Brian, et al.: "LibROSA: A Python Package for Music and Audio Analysis." Journal of Open Source Software - https://joss.theoj.org/papers/10.21105/joss.00534 (Дата обращения: 29.03.2024)

Thanks for attention!

Shkliar Mikhail Игоревич

Building a Music Recommendation System

mishklyar@edu.hse.ru