International Journal of Information Technology and Computer Science Applications (IJITCSA)
p-ISSN: 2964-3139 e-ISSN: 2985-5330
Vol. 02, No. 03, page 159 - 168
Submitted 21/09/2024; Accepted 20/11/2024; Published 27/11/2024

Optimizing User Engagement in Music Streaming Platforms: A Big Data Pipeline Case Study Using the Last.fm Dataset and the Hadoop Ecosystem

Akkord Ilgar Elizade
College of Computer and Information Sciences, Polytechnic University of the Philippines
e-mail : akkord.nsedfent@gmail.com

Corresponding Autor: Akkord Ilgar Elizade

Abstract
This paper proposed a big data pipeline to analyze user behavior on Last.fm which aims to make data-driven recommendations for improving user engagement and attracting new users. The comprehensive analysis of user behavior in the music streaming industry using the Hadoop ecosystem and data analytics techniques. Specifically, the study focuses on Last.fm, a popular music streaming platform that collects large amounts of user activity data. The article proposes a new data pipeline utilizing Hadoop Distributed File System (HDFS) for data storage and Apache Pig for data transformation, leading to improved data preprocessing and analysis. Various analyses are conducted, including identifying the most listened to artists, top users based on song consumption and social connections, artist popularity by tags, and the most recently tagged artists. The findings provide valuable insights into user preferences, current trends, and opportunities for enhancing the recommendation algorithm and user engagement. The article concludes by offering recommendations for personalized marketing strategies and curated playlists to increase user satisfaction and revenue.

Keywords: Hadoop ecosystem, HDFS, Apache Pig, big data, user behavior analysis

1 Introduction

The rapid advancement of technology has greatly impacted the music industry, transforming it from the traditional sale of physical music albums to the digital era of online music platforms. Today, music streaming platforms such as Spotify, Joox, YouTube Music, and Apple Music have revolutionized the way people consume music. Among these platforms is Last.fm, which started as an online radio station and evolved into a social network for music enthusiasts, offering personalized music recommendations and user-generated content [1].

Last.fm employs a subscription-based business model, where users can enjoy its services by paying a monthly fee or through advertisements. To enhance user satisfaction and engagement, Last.fm collects extensive data on user activities and behaviors, such as the songs they listen to, their favorite artists, and the tags they assign to artists. This data is utilized to optimize music recommendations and understand current music trends [2].

However, the analysis of this vast amount of data, known as big data, presents various challenges in terms of data integration, storage, preprocessing, and visualization. Last.fm currently faces limitations in its data pipeline, using a traditional relational database for data storage, which is not ideal for handling big data. Additionally, data preprocessing is performed only during the analysis and visualization stages using tools like Power BI, lacking a dedicated step for comprehensive data transformation.

In this article, we propose a new data pipeline for Last.fm that leverages the Hadoop ecosystem and data analytics techniques to effectively manage and analyze user behavior data. The proposed pipeline utilizes the Hadoop Distributed File System (HDFS) as the primary data storage solution, known for its scalability and fault tolerance. Apache Pig is employed for data transformation, enabling efficient processing of large datasets [3]. The data is then

loaded into Apache Hive for visualization and analysis, creating an integrated environment within the Hadoop ecosystem.

Furthermore, we explore the characteristics and features of the main datasets used for user behavior analysis, including artists, tags, user-artist interactions, user connections, and user-tagged artists with timestamps. These datasets are extracted from Last.fm using its API, providing valuable insights into user preferences and social network behavior [4].

To demonstrate the effectiveness of the proposed pipeline, we conducted several analyses on the Last.fm dataset. These analyses include identifying the most listened-to artists, top users based on song consumption and social connections, artist popularity by tags, and the most recently tagged artists. The findings from these analyses offer actionable insights for Last.fm, such as improving the recommendation algorithm, targeting influential users, and enhancing music discovery.

In conclusion, this article highlights the significance of leveraging the Hadoop ecosystem and data analytics in the music streaming industry, specifically focusing on Last.fm. The proposed data pipeline and user behavior analyses provide valuable tools for understanding user preferences, optimizing user engagement, and driving revenue growth. By embracing these techniques, music streaming platforms can effectively leverage big data to enhance user experiences and stay at the forefront of the evolving music landscape [5].

1.1 *General Business Process of Analyzing the Data*

The Last.fm dataset can be processed for analysis and data visualization in a general way by following a series of steps. These steps involve data extraction, preprocessing, transformation, analysis, and visualization [6]. The general process can be outlined as follows:

1. Data Extraction: The first step is to extract the relevant dataset from Last.fm's data sources. This can be done by utilizing Last.fm's API or accessing the dataset directly if it is publicly available. The datasets typically include information about artists, tags, user interactions, social connections, and timestamps.
2. Data Cleaning and Preprocessing: Once the dataset is obtained, it is necessary to clean and preprocess the data to ensure its quality and usability. This involves handling missing values, removing duplicates, standardizing data formats, and resolving any inconsistencies or errors in the dataset. Data preprocessing techniques such as data normalization or outlier detection may also be applied, depending on the specific requirements of the analysis.
3. Data Transformation: After preprocessing, the dataset may require further transformation to extract meaningful insights. This step involves performing calculations, aggregations, or applying statistical methods to derive relevant metrics or features from the data. For example, aggregating the number of songs listened to by each user for an artist or calculating the popularity of artists based on tag associations.
4. Analysis and Modeling: With the transformed dataset, various analysis techniques can be applied to uncover patterns, trends, and relationships within the data. This can include descriptive statistics, data mining algorithms, machine learning models, or network analysis methods. The specific analysis techniques used will depend on the research questions or objectives of the study.
5. Data Visualization: To effectively communicate the findings and insights from the analysis, data visualization plays a crucial role. Visualization techniques such as charts, graphs, heatmaps, or interactive dashboards can be employed to present the analyzed data in a visually appealing and informative manner. Visualization tools and libraries like Tableau, matplotlib, or D3.js can be utilized for this purpose.
6. Interpretation and Decision Making: The final step involves interpreting the results of the analysis and using them to inform decision-making processes. The insights gained from the analysis and visualization can be utilized to optimize user engagement strategies, improve music recommendations, identify emerging trends, target marketing campaigns, or enhance overall user experience on the Last.fm platform.

It is important to note that the specific details of each step may vary depending on the specific research goals, dataset characteristics, and analytical techniques employed. Additionally, the use of big data processing frameworks like Hadoop and Apache Spark can significantly enhance the scalability and efficiency of the analysis and visualization process [7].

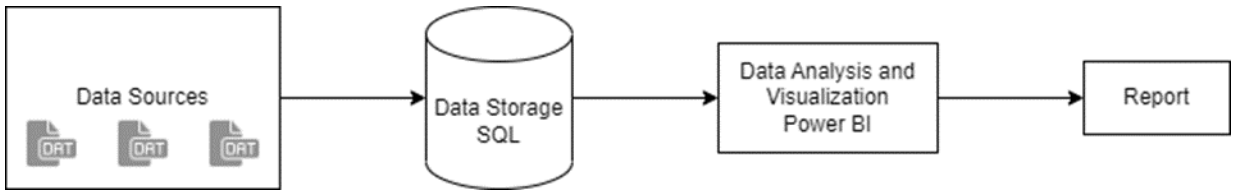

Figure 1. Current Data Pipeline of last.fm

Last.fm has been extracting their data source in dat format. These data are then stored in the data storage specifically the traditional relational database which is not suitable for handling big data due to its scalability of it. Furthermore, there is no proper data preprocessing step done for the data before it is stored inside the storage. The data will only be transformed once it comes to the analysis and visualization process using power BI to produce a static report. To analyze the users' behavior of last.fm, the proper data preprocessing should be done.

1.2 *Proposed Pipeline for Big Data Analysis*

After evaluating the current data pipeline used by last.fm, it is not an efficient pipeline to be used to perform user behavior analysis, thus the data pipeline should be designed in a way that matches the characteristics of big data and the goal of the analysis. It is important to ensure the quality of the data through the proper ETL process. Furthermore, data integration should also be applied to make sure all the data collected are in the uniformized format so it will be easier to conduct the analysis later [8].

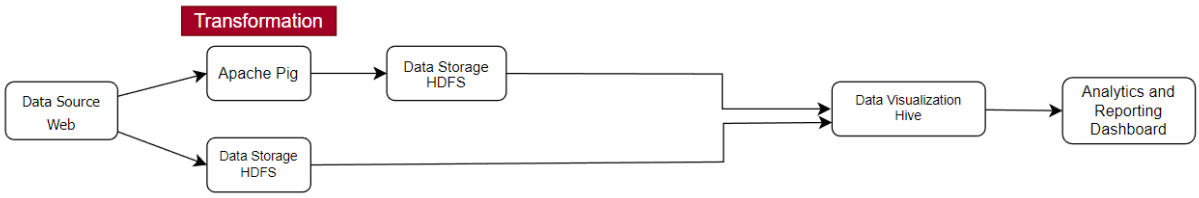

Figure 2. Proposed Data Pipeline for Last.fm

This paper has proposed a new data pipeline for last.fm to better manage the users' activity data that are retrieved so it can be used to provide better insight into users' behavior and help in the decision-making process. Here, the paper proposed to use Hadoop Distributed File System (HDFS) as the main data storage. HDFS is scalable and fault tolerant by using name node and data node as its main architecture, whereby the data is replicated across the nodes [9]. HDFS is widely used for handling big data due to its benefits. Subsequently, the transformation process will be done using Apache Pig. After that, the data will be stored in HDFS and loaded into Hive to proceed with the visualization. All of these processes will be done in one environment, which is the Hadoop ecosystem.

1.3 *ETL Process and Output*

As specified on the data pipeline, the ETL process will be done using Apache Pig which is one of the features in the HDP Sandbox. Apache Pig is one of the data analysis tools that can efficiently handle a large amount of data, it uses Pig Latin as the programming language [10]. This section will focus on describing the ETL process done for preparing last.fm's data before proceeding to the visualization part.

1. Data Extraction

   There are five (5) main datasets that are used for last.fm users' behavior analysis namely artists.dat, tags.dat, users_artists.dat, user_friends.dat, and user_taggedartists_timestamp.dat. These datasets were extracted from last.fm web using the API. Five (5) main data sets are used in this analysis, which are loaded into Apache Pig. The data is read into Apache Pig using the LOAD function.

```
--extract the data
artists = LOAD '/user/maria_dev/Assignment1/artists.dat' AS (id:int, name:chararray, url:chararray, pictureURL:chararray);
tag = LOAD '/user/maria_dev/Assignment1/tags.dat' AS (tagID:int, tagValue:chararray);
userArtist = LOAD '/user/maria_dev/Assignment1/user_artists.dat' AS (userID:int, artistID:int, weight:int);
userFriend = LOAD '/user/maria_dev/Assignment1/user_friends.dat' AS (userID:int, friendID:int);
userTagTimestamp = LOAD '/user/maria_dev/Assignment1/user_taggedartists-timestamps.dat' AS (userID:int, artistID:int, tagID:int, timestamp:long);
```

Figure 3. Data Extraction Script

2. Data Transformation

The transformation will generate 3 new datasets which are tags, users, and artists.

```
-- generate the Tags activities among users
getTags = FOREACH tag GENERATE tagID, tagValue;
taggingUsers = GROUP userTagTimestamp BY tagID;
tagCountByUsers = FOREACH taggingUsers GENERATE group AS tagID, COUNT(userTagTimestamp.userID) AS userCount;
tagging = JOIN tagCountByUsers BY tagID, getTags BY tagID;
tags = FOREACH tagging GENERATE tagCountByUsers::tagID, getTags::tagValue, tagCountByUsers::userCount;
--checking the result
DUMP tags;
```

```
-- generate the artist dataset
getArtistTag = GROUP userTaggedartist BY artistID;
countTheTag = FOREACH getArtistTag GENERATE group AS artistID, COUNT(userTaggedartist.userID) AS totalFan;
weightArtist = GROUP userArtist BY artistID;
-- find the totalListened for each artist
weightArtistCount = FOREACH weightArtist GENERATE group AS artistID, SUM(userArtist.weight) AS totalListened;
joinAll = JOIN countTheTag BY artistID, weightArtistCount BY artistID, artists BY id;
-- find the total tags associated with each artist
artist = FOREACH joinAll GENERATE countTheTag::artistID AS artistID, artists::name AS name, countTheTag::totalFan AS totalFan,
weightArtistCount::totalListened AS totalListened;
--checking the result
DUMP artist;
```

```
--generate the number of song each user listened
getUsers_all = FOREACH userArtist GENERATE userID;
--get only distinct users
getUsers = DISTINCT getUsers_all;
weightingUsers = GROUP userArtist BY userID;
usersWithMostListenedSong = FOREACH weightingUsers GENERATE group AS userID, SUM(userArtist.weight) AS totalListenedSongs;
compiledUsersSong = JOIN usersWithMostListenedSong BY userID, getUsers BY userID;
usersListenedToMostSong = FOREACH compiledUsersSong GENERATE usersWithMostListenedSong::userID, usersWithMostListenedSong::totalListenedSongs;
--find the number of friends for each user
getUsersFr_all = FOREACH userFriend GENERATE userID;
getUsersFr = DISTINCT getUsersFr_all;
friendUsers = GROUP userFriend BY userID;
usersWithMostFriends = FOREACH friendUsers GENERATE group AS userID, COUNT(userFriend.userID) AS totalFriends;
compiledUsersFriend = JOIN usersWithMostFriends by userID, getUsersFr BY userID;
usersMostFriends = FOREACH compiledUsersFriend GENERATE usersWithMostFriends::userID, usersWithMostFriends::totalFriends;
--join the number of songs listened and the number of friends into one table
users_compiled = JOIN usersListenedToMostSong BY userID, usersMostFriends BY userID;
users = FOREACH users_compiled GENERATE usersWithMostListenedSong::userID, usersWithMostListenedSong::totalListenedSongs, usersWithMostFriends::totalFriends;
--checking the result
DUMP users;
```

Figure 4. Data Transformation Script

In this process some basic transformation is done, here the tag dataset is reconstructed whereby one more variable for the user count is added. This user count is derived by aggregating the number of users who use the tag to tag the artist. This information is gotten from the user_taggedartists_timestamps.dat. Meanwhile, for the artist dataset, the artist ID, name, the total number of fans which is generated from user_taggedartists-timestamps.dat by aggregating how many times the tag is associated with each corresponding artist, and the total listened will be generated from the userArtist.dat by aggregating the total song listened based on each artist. Finally, the users' dataset will consist of the user ID, the total number of songs listened to which is generated from the userArtist.dat, and the total number of friends which is generated from userFriends.dat. After the data is transformed, the dataset will be stored in HDFS by using the code in Figure 5.

```
--store (load) to HDFS
STORE tags INTO 'output/Tags/' USING PigStorage;
STORE artist INTO 'output/Artist/' USING PigStorage;
STORE users INTO 'output/Users/' USING PigStorage;
```

Figure 5. Store Data to HDFS

Once the script is executed, the results can be seen from the results section like in Figure 6 whereby each tag is grouped together with the tag ID, tag value, and the number of users that use the tag. The same process is repeated for the other two datasets.

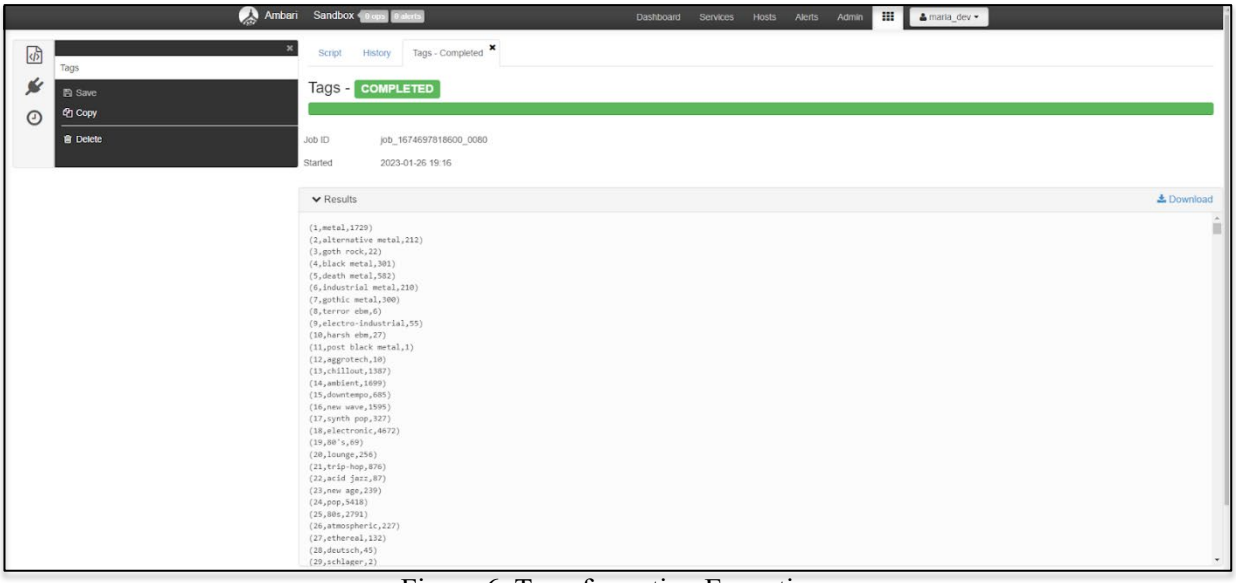

Figure 6. Transformation Execution

3. Data Loading

After the cleaned data has been stored in the HDFS it is then loaded to Apache Hive as shown in Figure 7, using the tab as the delimiter of the data. The results are shown in the last part of the figure whereby there are three columns in the dataset, tagID, tagValue, and userCount. These processes are also done for the other two datasets. The schema for each dataset is shown in Figure 8.

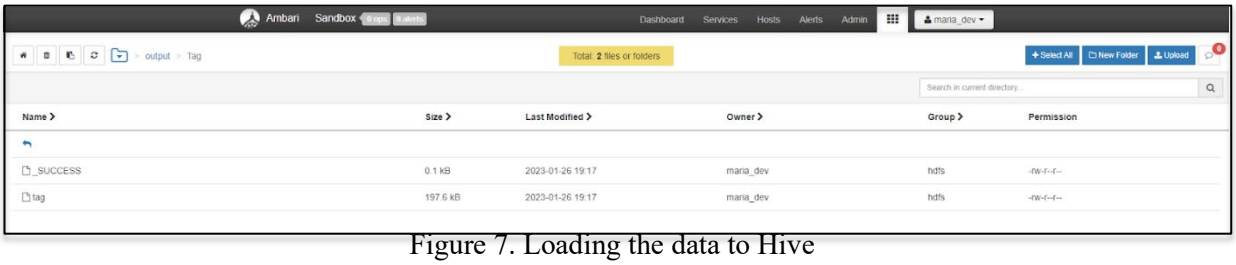

Figure 7. Loading the data to Hive

```
tags: {tagCountByUsers::tagID: int,getTags::tagValue:
chararray,tagCountByUsers::userCount: long}

artist: {artistID: int,name: chararray,totalFan: long,totalListened: long}
users:{usersListenedToMostSong::userID:int,usersListenedToMostSong::totalListe
nedSongs: long,usersMostFriends::totalFriends: long}
```

Figure 8. The Schema for each dataset

2 Users and Customers Behavior

Last.fm has been focusing on developing more user-generated content, and understanding users' behavior will help achieve the goal. User behavior analysis itself is one of the most critical aspects of understanding user needs and preferences, at the same time understanding current trends. In the case of last.fm, analyzing the users' behavior patterns can help it improve the recommendation algorithm that allows it to satisfy users with more accurate and relevant music suggestions [11]. Additionally, identifying trends such as emerging artists or songs will give valuable information to the music industry. In terms of social network behavior, last.fm can get more understanding on whether

or not the choices of music made by the user are actually affected by what their friends listen to. The social network of last.fm is made based on their musical taste meaning that most users make friends with those who have a similar musical taste with them [12].

User behavior analysis on music streaming platforms had an impact on improving the design and operation of the platform system. This paper will do five different analyses to extract patterns in the users' behavior by focusing on the most listened to artist, users who listened to the most songs, users with the most friends, popular artists by tag, and finally using timestamps to find the most recent artist tagged by the users.

2.1 *Most Listened Artist*

Analyzing the most listened to artists can give precious information for last.fm in several aspects. Firstly, identifying the most listened artist can also help to identify which genres are most popular among the users which is part of understanding the current trends. This information can be used to improve the music recommendation system by providing users with their current favorite artist and genre. The better the recommendation system, the more engagement and retention of the users can be improved which also led to an increase in the revenue from subscription fees as well as ad's view. Additionally, understanding the most listened-to artist can also help in the platform marketing strategy, whereby the last.fm can use the most listened-to artist as the brand ambassador for the platforms to increase the number of users, besides it can create a personalized playlist for this particular artist to attract their fans to use and subscribe last.fm.

To analyze the most listened artist the main metric will be the total listened for each artist. This will be retrieved from the artist table; the data will be filtered to the top 20 only and sorted in descending order by totalListened.

```
SELECT TOP 20 artistID, name, totalListened FROM artist ORDER BY totalListened DESC;
```

Figure 9. SQL Script to get Most Listened to Artist

2.2 *Top Five Users based on the Most Listened Songs*

Similarly analyzing the top users based on the total listened to songs can provide insight into several aspects. Firstly, it will help to identify those top-tier users who are actually attached to the platform, when the users have a high total listened to songs, it is indicating that the users spend a lot of time using last.fm. These users can be the best target to offer special promotions to make sure they continue using the platform as well as share their preferences with their friends to also increase the number of new users and increase the revenue. Last.fm can also use the behavior of the top five (5) users to develop a targeted marketing strategy to target new users with similar preferences.

Analyzing the user who listened to the most songs can be done by using the totalListenedSongs in the user's table. These data will be filtered only for the top five (5) rows and sorted in descending order.

```
SELECT TOP 5 userID, totalListenedSongs FROM users ORDER BY totalListenedSongs DESC;
```

Figure 10. SQL Script to get Top Five Users based on Total Songs Listened

2.3 *Top Five Users with Most Friends*

Analyzing the top five (5) users with the most friends will allow last.fm to understand the social network of the users which helps to identify potential influential users who have a great number of connections. Last.fm then can approach these users through a targeted marketing campaign and special promotions so they can reach wider audience groups to introduce last.fm and attract new users.

This analysis can be done by using the total number of friends each user has, which has been calculated and stored in the users' table. Similar to the previous one, the data will be filtered to only include the top five (5) rows and sorted in descending order.

```
SELECT TOP 5 userID, totalFriends FROM users ORDER BY totalFriends DESC;
```

Figure 11. SQL Script to get Top Five Users with the Most Number of Friends

2.4 *Artist Popularity by Tag*

Analyzing the artist's popularity by the tag can help last.fm to improve music discovery, when the artists have more tags, it means they are more likely preferred by a larger number of users which makes them deserve to be at the top of the discovery page. When the users look at the discovery page, they can effortlessly listen to the music from their favorite artists who at the same time can increase user engagement. Finally, it is also used to analyze the current trend so last.fm can understand which artists are becoming more popular among the users and this information can be used by last.fm to strategize their content so it is always up to the trends.

This analysis can be done by using the total fans from the artist table. This total number of fans was derived from the number of tags associated with each artist. The data is sorted in descending order and filtered to only the top 20 rows.

```
SELECT TOP 20 name, totalFan FROM artist ORDER BY totalFan DESC;
```

Figure 12. SQL Script to get Top 20 Fans

2.5 *Five Artists with the Most Recent Tag*

When analyzing the five artists with the most recent tag, last.fm can gain a real-time insight into which artists are currently popular among the users based on their tags. This real-time insight can make sure that last.fm always keeps up with the current trend and can provide a more accurate user recommendation system. At the same time, this insight can also be used to improve music discovery content for the users.

To analyze the top five artists with the most recent tag, the timestamp is needed. However, in the cleaned dataset there is no timestamp generated, thus the timestamp will be retrieved from the user_taggedartists-timestamp.dat. Since the analysis wants to focus on the artist, knowing the ID won't be sufficient, hence the artist's name will also be retrieved by joining the user_taggedartist-timestamp.dat with the artist table by using the artistID. First, the user_taggedartist-timestamp.dat data will be loaded to the hive as taggedartist table, then the left join will be done by using artistID. The left join is done to make sure that all the rows from the taggedartist table are included in the join.

```
SELECT TOP 5 name, timestamp FROM taggedartist LEFT JOIN artist ON taggedartist.artistID =
artist.artistID ORDER BY timestamp DESC;
```

Figure 13. SQL Script for Top Five Artists with the Most Recent Tag

3 Findings

3.1 *Business Insight Based on the Analysis*

After conducting the analysis, there are several insights that can be highlighted.

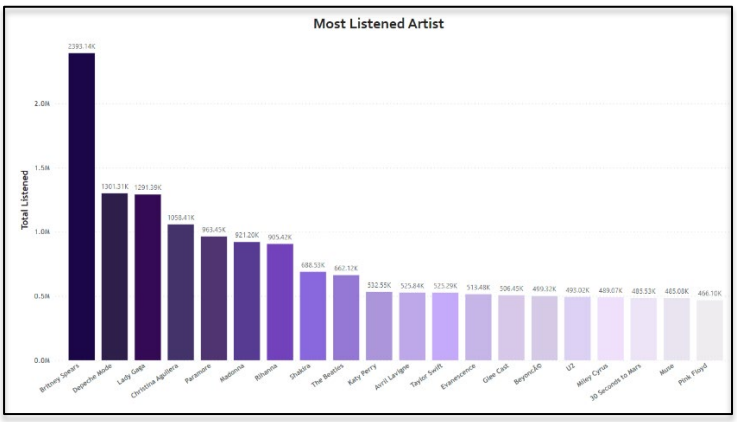

Figure 14. Most Listened Artist

According to Figure 14, the top three (3) most listened artists are Britney Spears with almost 2.4 million total listened, followed by Depeche Mode with a total listened around 1,3 million, and Lady Gaga with a total listened of approximately 1.29 million. This analysis shows which artists are famous among the users and it can be roughly seen that these artists came from the pop genre. Last.fm can use this information to create personalized playlists to attract users. Furthermore, this information can also be used to consider artist partnerships for any marketing campaign or brand ambassador. All of this can also lead to an increase in user engagement and revenue.

Figure 15 highlights the top five (5) users according to the total songs listened which are indicated by the user ID. User 757, 2000, 1418, 1642, and 1094 are the top five (5) users who listened to the most songs with the highest number of songs listened is 480k. This indicates that these five (5) users are the most active users on the platform and highly engaged with last.fm. Last.fm can approach these users to give them targeted promotions as an incentive for their loyalty to use the platform. This can also increase these users' satisfaction and keep their engagement to last.fm.

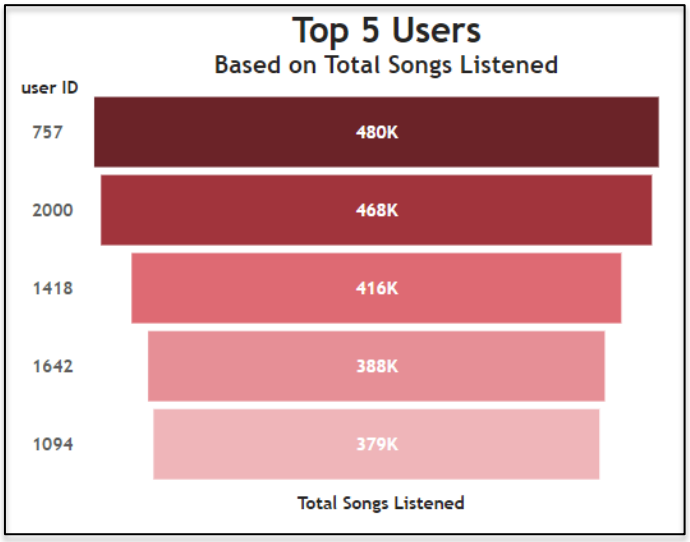

Figure 15. Top 5 Users Based on Total Songs Listened

Next, Figure 16 lists the top five (5) users based on the number of friends or connections they have. It can be seen that users 1543, 1281, 831, 179, and 1503 are users who have the largest connection on the platform, these users might have a higher influence in the community and have the potential to reach larger scope of audiences. Firstly, other users might be influenced by their music preferences and so on, so last.fm might approach them to help in the marketing campaign.

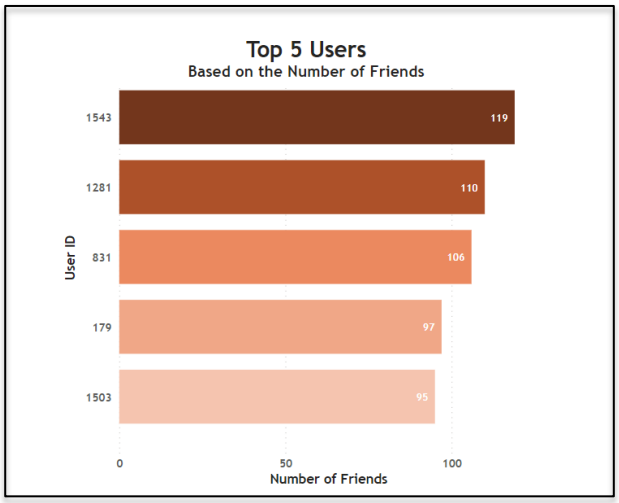

Figure 16. Top 5 Users Based on the Number of Friends

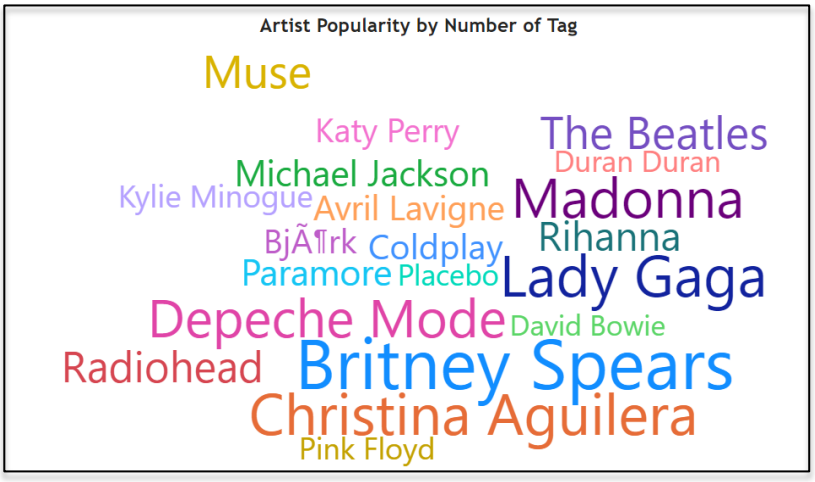

Figure 17. Artist Popularity by Number of Tag

Similar to the first analysis, Figure 17 highlights the artist's popularity by the number of tags they are associated with. Notice that the artist's popularity reflected by the number of tags shows similar results with those reflected by the total listened with Britney Spears as the most famous artist. Similar to the most listened to artist, this analysis can help last.fm to know which artists are more appealing to the users and similarly can be applied to design the personalized playlist and any other marketing campaigns which can help to boost user engagement and revenue.

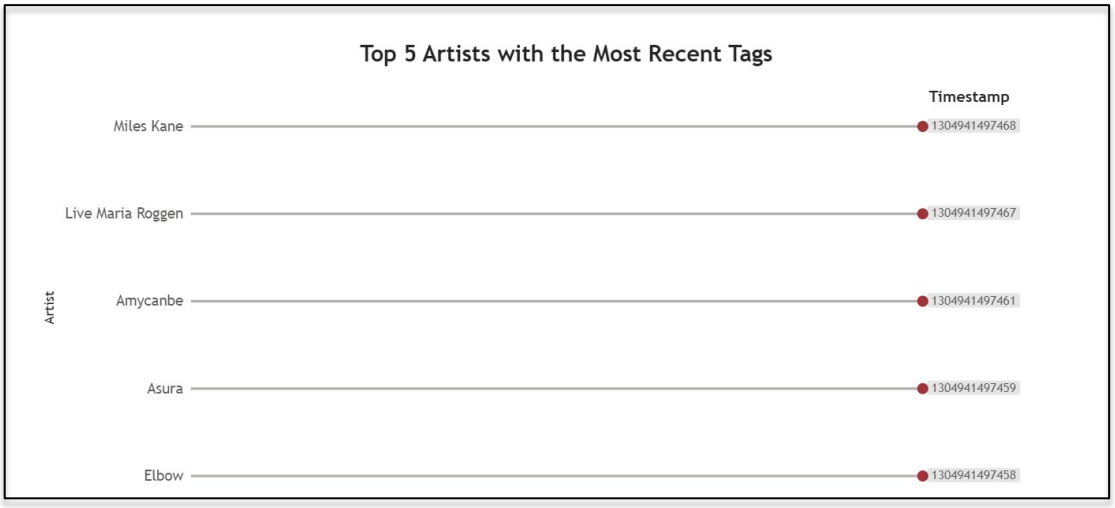

Figure 18. Top 5 Artists with the Most Recent Tags

This analysis can give last.fm a brief overview on the current trend and shows that the most recently tagged artists are Miles Kane, Live Maria Roggen, Amycanbe, Asura, and Elbow. When the artist is recently tagged, he or she might have just released a new music or rising artist who has been featured in some big project. This information can help to promote these rising artists and feature them in the new music or rising artist playlists which can also improve the music discovery process on the platform. When the platform is up to date, there will be more users interested in joining the community which will increase user engagement.

3.2 *Recommendation*

From the user behavior analyses that have been done, there are several recommendations that can be derived for the business to increase user engagement. Firstly, last.fm can consider doing partnerships with the most famous artist according to the most listened artist and artist popularity by tag. The partnership can be in terms of marketing projects or campaigns which target these artists' fans to become users of last.fm. Secondly, as highlighted in the findings part, curated playlists can be developed based on the current famous artist or new artists. This can improve the users' experience and make the music discovery process easier.

Next, last.fm can also consider more personalized marketing strategies for the users, such as for those top users who are active on the platform to give them some sort of rewards like subscription discounts or trials to make them try different subscription plans so they can consider upgrading their subscription plan. Similarly, a marketing campaign to invite new users using referral codes to get some discounts can be targeted to those users with the greatest number of friends. By using this marketing strategy, last.fm can get more users and at the same time can maintain the current users' engagement.

4 Conclusions

This paper proposed a new data pipeline to process big data specifically user behavior data from last.fm. This paper started by discussing the aim of the analysis which is to increase user engagement and revenue. The paper finds that the most listened artists and most popular artists based on the number of tags are predominantly from the pop genre which can be noted to improve music discovery. Then, the paper also highlights the top five (5) users based on the total listened songs are those who have a higher engagement to the platform and can be targeted for special marketing campaigns. Additionally, the marketing teams can also focus on the top five (5) users with the most friends listed in this paper to help the platform increase the number of new users. Finally, this paper also found the top five (5) artists with the most recent tag which might be used to understand the current new rising artist and be used to improve the music discovery process. Overall, these analyses can be used to address the goal of the analysis which is to increase user engagement and revenue for last.fm.

All of these analyses are done by following the proposed pipeline and since the paper has fulfilled the final goal, then it can be concluded that the proposed data pipeline worked well with the current dataset given and might also be used to do different types of analysis with different goals. Additionally, using one environment to do all the processes has also increased the efficiency of the pipeline.

BIBLIOGRAPHY

[1]. "Mining user generated data for music information retrieval," Mining User Generated Content, pp. 107–136, Jan. 2014. doi:10.1201/b16413-14.

[2]. S. Sela, Improvised music follows human language quantitative properties to optimize music processing, Dec. 2021. doi:10.31234/osf.io/fh4qu.

[3]. C. S. R. Prabhu, A. S. Chivukula, A. Mogadala, R. Ghosh, and L. M. J. Livingston, "Big Data Tools—hadoop ecosystem, Spark and NoSQL databases," Big Data Analytics: Systems, Algorithms, Applications, pp. 83–165, 2019. doi:10.1007/978-981-15-0094-7_4.

[4]. N. Gerhart and M. Koohikamali, "Social Network Migration and anonymity expectations: What anonymous social network apps offer," Computers in Human Behavior, vol. 95, pp. 101–113, Jun. 2019. doi:10.1016/j.chb.2019.01.030.

[5]. Y. M. Kassa, R. Cuevas, and A. Cuevas, "A large-scale analysis of Facebook's user-base and user engagement growth," IEEE Access, vol. 6, pp. 78881–78891, 2018. doi:10.1109/access.2018.2885458.

[6]. H. Qin et al., "Building Electricity Consumption Analysis: Data-driven approach with preprocessing, visualization, and cluster analysis," 2023 International Conference on Electronics and Devices, Computational Science (ICEDCS), pp. 48–53, Sep. 2023. doi:10.1109/icedcs60513.2023.00015.

[7]. E. Nazari, M. H. Shahriari, and H. Tabesh, "Bigdata analysis in Healthcare: Apache Hadoop , Apache Spark and Apache Flink," Frontiers in Health Informatics, vol. 8, no. 1, p. 14, Jul. 2019. doi:10.30699/fhi.v8i1.180.

[8]. Jala Aghazada, "Arrangement and modulation of ETL process in the storage," Science Review, no. 1(28), pp. 3–8, Jan. 2020. doi:10.31435/rsglobal_sr/31012020/6866.

[9]. A. D. Jadhav and V. Pellakuri, "Accuracy based fault tolerant two phase - intrusion detection system (TP-IDS) using machine learning and HDFS," Revue d'Intelligence Artificielle, vol. 35, no. 5, pp. 359–366, Oct. 2021. doi:10.18280/ria.350501.

[10]. B. Vaddeman, "Pig Latin in Hue," Beginning Apache Pig, pp. 115–122, 2016. doi:10.1007/978-1-4842-2337-6_8.

[11]. Z. Liu and F. Ren, "Algorithm improvement of movie recommendation system based on hybrid recommendation algorithm," Frontiers in Computing and Intelligent Systems, vol. 3, no. 3, pp. 113–117, May 2023. doi:10.54097/fcis.v3i3.8581.

[12]. J. I. Criado and J. Villodre, "Public employees in social media communities: Exploring factors for internal collaboration using social network analysis," First Monday, Apr. 2018. doi:10.5210/fm.v23i4.8348.