Music Recommendations using Million Song Dataset

Sai Prasanna Kumar Kumaru
MS Artificial Intelligence
San Jose, US
saiprasannakumar.kumaru@sjsu.edu

Sai Teja Kandukuri
MS Artificial Intelligence
San Jose, US
saiteja.kandukuri@sjsu.edu

Neeharika Yeluri
MS Artificial Intelligence
San Jose, US
neeharika.yeluri@sjsu.edu

***Abstract*—This project aims to develop a music recommendation system using the Million Song Dataset, specifically the Taste profile dataset. The dataset consists of real user-song-play count triplets, artist and song metadata such as artist name, title, artist hotness, year of release, duration, etc., The project involves tasks such as data preprocessing, exploratory data analysis, feature engineering, and building and evaluating different recommendation algorithms such as Neural Collaborative Filtering, hybrid recommender system using autoencoders, and Singular Value Decomposition (SVD) with neural network-based approach. The project also explores incorporating additional metadata such as artist and song information to enhance the recommendation quality. This paper utilized lyrics data, including word count, to gain insights and train a neural network model for predicting artist familiarity based on song metadata, such as hotness, duration, and year of release. The performance of different models is evaluated using common evaluation metrics such as MSE, RMSE.**

***Index Terms*—Neural Network, AutoEncoders, Collaborative Filtering, Recommender system, Million Song Dataset(MSD)**

I. Introduction

“There is no data like more data” [1]. The popularity of music streaming services has grown over the past few years, creating a sizable volume of user-generated data.This information can be used to create effective music recommendation systems that can offer users individualized and pertinent music selections. Systems for making music recommendations not only improve user experience but also boost earnings for music streaming firms. Therefore, it is crucial for music streaming services to develop an accurate and effective music recommendation system. The Million Song Dataset (MSD) is a large scale dataset. “The MSD contains metadata and audio analysis for a million songs that were legally available to The Echo Nest.” [1]. We picked a subset of the dataset called ‘Taste profile dataset’ and applied a few algorithms using metadata.

A. Motivation

The explosive growth of digital music services and the increasing number of music consumers worldwide has led to a tremendous amount of user-generated data, which can be leveraged to build powerful music recommendation systems. Music recommendation systems help music streaming platforms provide a personalized and relevant music experience to their users. This not only enhances user engagement and retention but also increases revenue for music streaming services. Therefore, building an accurate and efficient music recommendation system is of significant importance to music streaming companies. The project aims to provide a comprehensive analysis of different algorithms and techniques used for music recommendation and their effectiveness. The ultimate goal is to build a music recommendation system that can provide a personalized and satisfying music experience to users.

B. Dataset Description

The Million Song Dataset (MSD) is a large-scale music dataset that contains audio features and metadata for one million popular songs. It is created by The Echo Nest and available for research and non-commercial use [1]. The dataset includes 48,373 unique songs, 1,019,318 unique users, and 48,373,586 listening events. The user-song interaction data is a subset of the main dataset that contains real user, song, and play count triplets and can be primarily used for collaborative filtering [1]. This subset contains features such as artist name, song name, user request, and play count. The MSD also includes additional metadata for each song, including artist, title, tempo, key, loudness, and many other audio features extracted using music information retrieval (MIR) techniques [1].

The MSD also contains a dataset of song lyrics and artist biographies, called the MusiXmatch dataset, which can be linked to the MSD using song and artist identifiers [2]. The MusiXmatch dataset contains over one million song lyrics and metadata, including information such as artist, album, language, and lyrics themselves. The dataset is made available for research purposes only and can be used in combination with the MSD to build more comprehensive music recommendation systems that take into account song lyrics and artist biographies [2].

II. RELATED WORK

In the field of music recommendation systems, one important task is cover song detection. This task has traditionally been addressed using audio-based systems, but the performance of such systems has been limited and their scalability for large-scale applications has been a challenge. A study conducted by Albin et al.[3] on Million Song Dataset (MSD) investigated the use of text-based similarity measures in addition to state-of-the-art audio similarity measures to improve the accuracy and scalability of cover detection systems. The results of the study showed that the use of standard tf-idf based text similarity measures on song titles and lyrics can significantly improve the accuracy of cover detection systems on MSD. “Specifically, the work achieved a 35.5% absolute increase in mean average precision compared to the previous state-of-the-art scalable audio content-based methods.” [3].

Various approaches have been proposed for addressing the problem of sparse data, especially implicit feedback data. A study conducted vy Harald[4] on the Million Song Dataset (MSD) evaluated the performance of a linear model for recommender systems on publicly available datasets. The study found that this model achieved better ranking accuracy than various deep non-linear models and state-of-the-art collaborative-filtering approaches, including neighborhood-based approaches, which have been found to outperform model-based approaches in the Million Song Dataset. Moreover, the study showed that the proposed linear model outperformed the best competing model by a significant margin, achieving improvements of 25% in Recall on the MSD data.

Recently, there is a growing interest in exploring non-linear probabilistic models that can overcome the limitations of linear factor models in collaborative filtering research. One such model is the variational autoencoder (VAE) that has been extended to collaborative filtering for implicit feedback. Dawen et al. [6] introduces a generative model with a multinomial likelihood and employs Bayesian inference for parameter estimation. ”While the multinomial likelihood is widely used in language modeling and economics, it receives less attention in the recommender systems literature. Empirically, the study shows that the proposed approach outperforms several previous studies on MSD.

Janne et al.[5] introduce a self-supervised, contrastive learning approach for musical representation learning called CLMR, which uses SimCLR and a large chain of audio data augmentations. This method is able to learn useful representations without the need for labels. “The paper demonstrates that a linear classifier trained on the proposed representations achieves higher average precision than supervised models on the Million Song dataset. The authors also show that CLMR’s representations are transferable to out-of-domain datasets, indicating strong generalizability in music classification. Finally, the authors demonstrate that their approach allows for data-efficient learning on smaller labeled datasets.” [5].

Jordi et al.[7] developed models that rely on domain knowledge, such as log-mel spectrograms with a convolutional neural network designed to learn timbral and temporal features. The results suggest that music domain assumptions are relevant when not enough training data are available, and waveform-based models outperform spectrogram-based ones in large-scale data scenarios. The spectrogram model performed well indicating that musical knowledge can be useful for designing models for the MSD.

III. Design and Implementation

Experiments are conducted on taste profile dataset in million song dataset(MSD). We performed to evaluate the different recommender systems and trained three different algorithms Support Vector Machines and Neural Network-based approach, Neural collaborative filtering, and Hybrid Recommendation using auto-encoders. We benchmarked the above models on basic algorithms like SVD, NN without metadata.

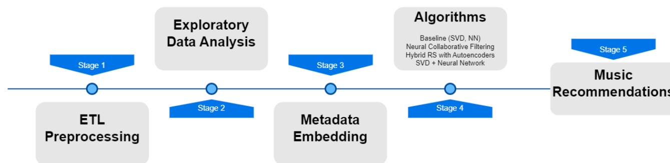

Fig. 1. Model Pipeline.

A. Data Preprocessing

In the data preprocessing stage, two types of data were dealt with: interaction data and metadata. Outliers in the user song play counts of interaction data were removed using interquantile range. The play counts ranged from 1 to 920, but mostly from 1 to 6, so the outliers were removed to keep the play counts within the range of 1 to 6. In the metadata, irrelevant columns such as artist mbid, track 7 digit id, shs perf, shs work, etc. were removed. NaN values were also removed. Text columns such as release, title, and artist name were preprocessed by removing punctuation marks.

Lyrics data - The dataset is available in .db format and includes a word count for each song.

B. Exploratory Data Analysis

Most popular songs, artists, average song duration, release year, distribution of artist familiarity, artist hotness. correlation matrix between familiarity, hotness and word count.word count, on an average 191 and unique words are 74.

The Taste Profile dataset is a subset of the Million Song Dataset that contains user-id, song-id, and play count information. It has 2 million rows and 3 columns and does not have any missing or duplicate values. The Metadata table and track metadata include information about songs such as artist familiarity, artist hotttnesss, artist location, genre, release, song hotttnesss, song id, and title. Some columns have missing values and duplicates have been removed to reduce dimensionality. The two datasets are merged into a new dataframe with over 1 million rows and 11 columns. The merged dataframe is grouped by song title, and the top 10 songs with the highest play counts are selected, with ”Use Somebody” being the most popular song. The play counts are then aggregated by artist name, and the top 10 artists are generated with Kings Of Leon, Coldplay, and Muse being the top 3.

The below figure 2 is a histogram of play counts based on users in the merged data. After grouping the merged data

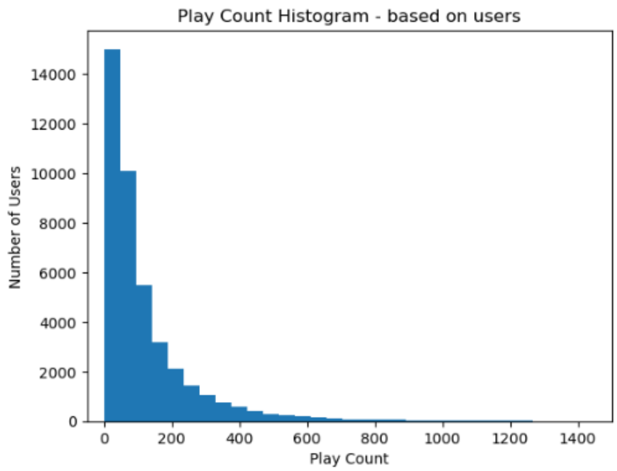

Fig. 2. Play Count distribution.

by user id it calculates the sum of play count for each user. This resulting histogram shows the distribution of play counts among users, with the x-axis representing the play count and the y-axis representing the number of users that have that play count. The histogram shows that it is mostly right skewed and the majority of users have low play counts, with a long tail of users with higher play counts.

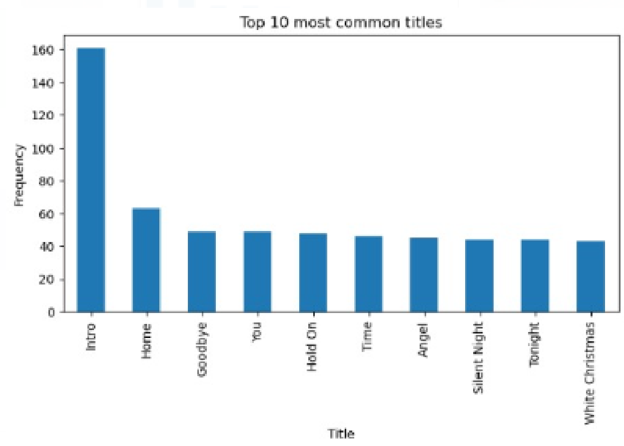

Fig. 3. Common titles in song metadata

According to the above histogram of song metadata most of the 10 common titles are Intro, Home, Goodbye, You, Hold On, Time, Angel, Silent Night, Tonight, and White Christmas.

The below scatter plot shows the relationship between artist hotttnesss and artist familiarity variables in the song data dataframe. The plot is a scatter plot, where each point represents an artist. The x-axis represents the artist hotttnesss variable, which is a measure of how popular an artist is. The y-axis represents the artist familiarity variable, which is a measure of how well-known an artist is. It can be inferred from the plot that there is a positive correlation between the

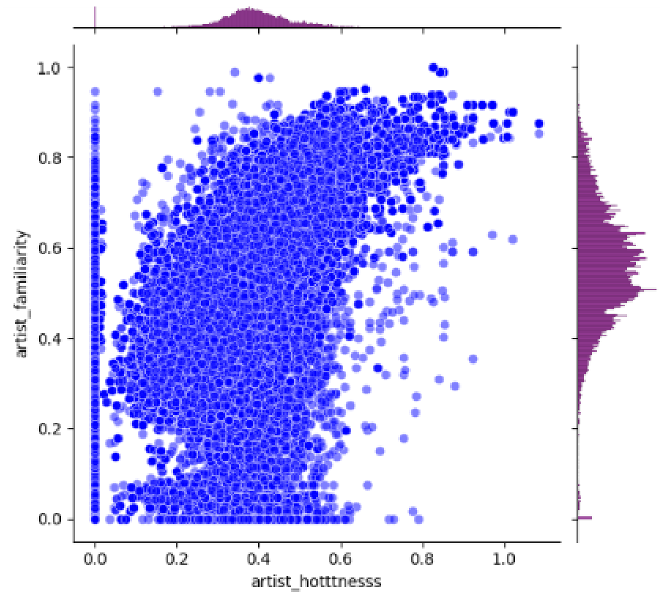

Fig. 4. Correlation plot

two variables, indicating that popular artists tend to be more well-known.

C. Algorithms

1) Baseline Models: We performed analysis on baseline models SVD, Basic Neural Network without metadata to test our model performances on a subset of Taste Profile Dataset.

2) Neural Collaborative Filtering: Neural Collaborative Filtering (NCF) is a deep learning-based approach to collaborative filtering that has gained popularity in recent years due to its effectiveness in recommendation tasks. Collaborative filtering is a widely used technique in recommendation systems, which recommends items based on the user's past interactions and the interactions of similar users. NCF incorporates both the user's and the item's embeddings in a neural network-based model to predict the user's rating or preference for a particular item. This approach can model complex user-item interactions, including non-linearities and high-order interactions. The model is typically trained using a large dataset of user-item interactions and can be optimized using various loss functions such as mean squared error or binary cross-entropy based on output variable type(For example, if play count as output variable then we use mean squared error else if play count is normalized as shown below.

$$\text{song likeness} = \frac{\text{play count of the song listened by a user}}{\text{total song play counts of the user}} \tag{1}$$

We have implemented a model based on a collaborative filtering approach, where we try to predict the recommendations of songs of a user by using play count and likeness of song separately. Additionally, we have also used metadata information about songs such as year, artist, title and release to improve the recommendations.The first step in the

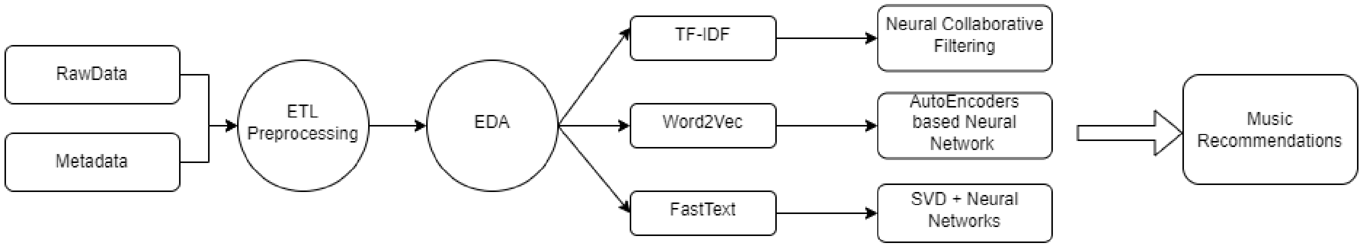

Fig. 5. Model Pipeline.

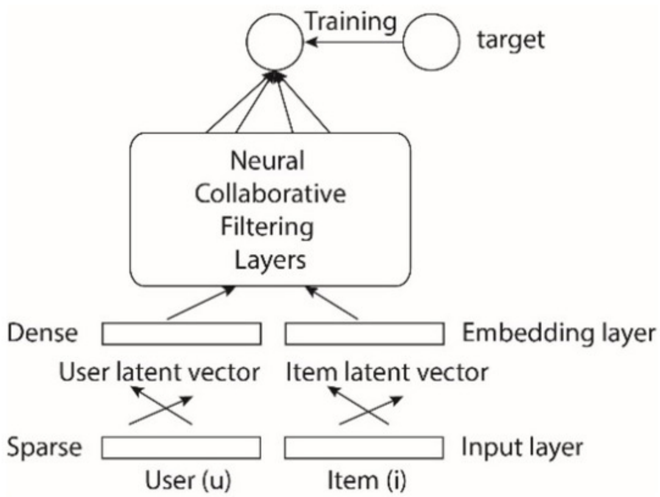

Fig. 6. Model Pipeline.

implementation was data preprocessing, we have used two datasets, one containing user play counts for various songs and the other containing metadata information about those songs. The main dataset contained 3 columns - user id, song id, and playcount, and the metadata dataset contained 5 columns - title, release, artist name, duration, artist familiarity, artist hotttnesss and year. We have merged these two datasets on song id and removed any duplicate records. For our analysis, we have converted the text columns of our data, such as artist name, title, and release, into word embeddings using the tf-idf vectorizer. These embeddings are then merged with the main dataframe based on the song ID. As a result, our final input variables consist of the song ID, user ID, and metadata vector with embeddings. This allows us to utilize all of these variables in our analysis and make predictions based on a combination of song and user data.

The model architecture consists of three main components - embedding layers for songs and users, a dense layer for metadata, and fully connected layers for feature extraction and prediction. We have used an embedding layer to learn a low-dimensional representation of songs and users, which captures their inherent characteristics. We have also used a dense layer for metadata to extract meaningful features from the metadata information. We have then concatenated the outputs of all three layers and passed them through fully connected layers to make the final prediction. Then the model was trained on the preprocessed data using the Adam optimizer with a learning rate of 0.001 and a batch size of 64. We have used Mean Squared Error (MSE) as the loss function and Root Mean Squared Error (RMSE) as the evaluation metric. We have also used early stopping to prevent overfitting and validate the model's performance. The model was trained for 20 epochs, and the training process was monitored for validation loss.

3) Hybrid Recommendation using AutoEncoders: Hybrid Recommendation using AutoEncoders is a state-of-the-art approach to building recommendation systems that leverage both content-based and collaborative filtering techniques. The approach involves encoding user and item features using deep neural networks, which are then used to generate recommendations. One of the key advantages of this approach is that it can handle sparse data more effectively than traditional collaborative filtering techniques. Additionally, it can capture complex, non-linear relationships between user and item features that are often missed by linear models. The model utilizes both interaction data (user-song-playcount triplet) and metadata to provide personalized recommendations to users. The dataset used includes play counts and metadata that includes release, title, artist names, artist hotness, artist familiarity, and duration of the song, and year of song releaseText columns from the metadata are passed through a Word2Vec model to generate embedding vectors. Later, numerical columns are attached to form an array with all metadata information.

The network architecture for the hybrid model includes separate input layers for interaction data and metadata. Embedding layers are defined for both input layers to generate embedding output. An autoencoder is used for the metadata input layer, where dense layers are defined for encoding and decoding metadata information. The encoded layer is merged with the embedding output of interaction data, and the output of the merged layer is passed through several dense layers with dropout regularization. Finally, the output layer is defined with a single neuron to predict the rating.

The full model is compiled with the Adam optimizer with a learning rate of 0.001, mean squared error (MSE) as a loss function, and mean absolute error (MAE) as a metric. The output of the model represents the predicted rating, which can be used to recommend songs to the user. This hybrid model combines the strengths of collaborative filtering and content-based filtering and provides personalized recommendations to users by considering both user-item interactions and item

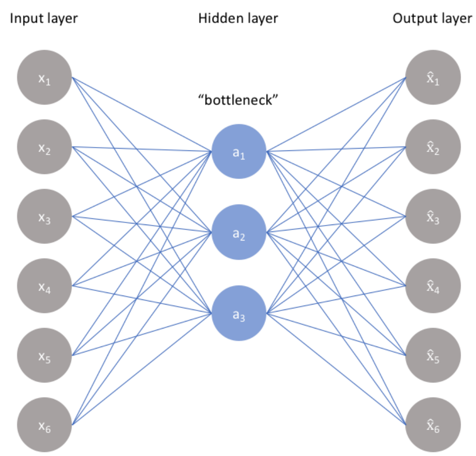

Fig. 7. Model Pipeline.

metadata information.

4) Singular Value Decomposition and Neural Network-based approach: Singular Value Decomposition (SVD) is a popular matrix factorization approach that has several applications in various fields, including natural language processing, collaborative filtering-based recommendation systems, and image processing. SVD can be used to find latent factors or features that explain the user-item interactions in a given matrix, typically consisting of user-item ratings. The original matrix is factorized into the user latent factor matrix and item latent factor matrix. The SVD model can learn the underlying patterns from the data to reconstruct the original matrix and make predictions for the missing play counts, which can then be used to generate song recommendations. However, it might not be able to capture complex patterns and interactions among users, items, and other factors in the data. To address this limitation, SVD's predictions can be combined with other features or meta-data and trained with a neural network to improve recommendations quality. This approach allows the neural network to learn more representations of user-item interactions and also learn from the metadata, resulting in improved prediction accuracy. Overall, in the context of music recommendation systems, SVD can be combined with features such as song metadata or user listening history, and fed into a neural network to generate personalized and accurate music recommendations.

Initially, we trained the SVD model on user, song, and play count triplet pairs. To find the best parameters for SVD, the GridSearch technique is used. SVD is trained on 2 n factors, with 0.007 learning rate and 0.1 regularization. Predictions obtained from SVD are stored for further use as inputs for the neural network. For the neural network, FastText embeddings of 30 dimensions are obtained for all the text columns. Textual representations from FastText, numerical features such as artist familiarity, artist hotness, song hotness, song duration, and predictions obtained from SVD are fed into the neural network model and are passed as metadata to the neural network. The network architecture has user input, song input, and metadata input. Embeddings are computed for users and items and concatenated with metadata embeddings. The neural network consists of several dense layers with relu activation and finally, a custom activation function is used to generate predictions.

The neural network model is compiled with a 0.001 learning rate using the Adam optimizer with RMSE and accuracy as metrics. Loss, accuracy, and RMSE curves are plotted to evaluate the model's performance. The play count predictions obtained from this model are used to recommend songs to the users. The combination of SVD and neural networks to build the recommender system has shown promising results and outperformed baseline architectures. By using the predicted ratings from SVD as additional features for a neural network, the performance of the system was improved. This approach allows us to leverage the strengths of both techniques and achieve a more effective music recommendation system that can provide a better user experience. Overall, this SVD and neural network-based hybrid approach is a powerful tool to generate personalized music recommendations to users.

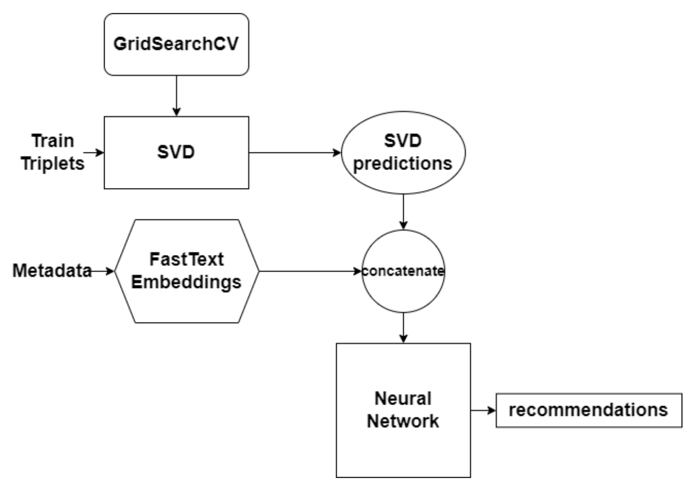

Fig. 8. Model Pipeline.

5) Artist Familiarity Prediction: In this experiment, we are exploring the relationship between artist popularity and various song metadata. We started by preprocessing the lyrics data which contains track ID, MXM track, and word count. We also have metadata which contains all song-related columns such as artist name, release year, title, etc.

To perform exploratory data analysis, we analyzed the lyrics data and found the top 10 common words, the distribution of word count in songs, and the correlation between total words and total unique words. We also analyzed the correlation between artist hotness and artist familiarity and created word clouds for the title, release, and artist name to visualize the most common words in each of them.

Next, we combined all the words for each lyric and preprocessed them using the NLP package NLTK by removing stop words, punctuations, etc. Then we split the data into a train set (80%) and a test set (20%). Our target variable is artist hotness, and our input variables contain all song metadata and artist familiarity. We trained a neural network model with auxiliary input and output using Keras, which helps us to test whether artist familiarity has any impact on artist hotness or popularity.

The neural network model architecture involves two input layers: input 1 which represents the total words and lyrics vector and input 2 which represents artist hotness. We define an embedding layer for input 1, followed by a flatten layer, two dense layers, and a dense layer for input 2. Then we concatenate input 1 and input 2 to create a single vector, which is then passed through two output layers: aux output and main output. The model is trained using Adam optimizer, binary cross-entropy loss, and mean squared error (MSE) as the metric.

We want to test whether the model can predict artist hotness accurately and whether artist familiarity has any impact on artist popularity. To evaluate the model, we check the root mean squared error (RMSE) and mean absolute error (MAE) after training. We found that the model is able to predict artist hotness reasonably well even without considering artist familiarity, but adding artist familiarity gives more accurate information about artist popularity.

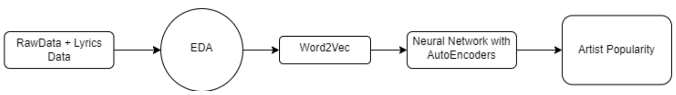

Fig. 9. Music Recommendation for Users

D. Technologies & Tools used:

The Keras Deep Learning library was utilized to train Neural networks and the surprise library was used to test the data on baseline models. Additionally, scikit learn and matplotlib were employed for data preprocessing and visualization, while the gensim package was used for the word2vec model. We used Fasttext, Word2vec, tfidf packages to generate word embeddings.

IV. Experiments and Results

TABLE I
RMSE and MAE for Algorithms below on play count

Algorithm	RMSE	MAE
SVD	1.21	0.87
Basic Neural Network(without metadata)	1.23	0.93
Neural Collaborative Filtering	1.8	1.3
AutoEncoders based Neural Network	1.20	0.86
SVD + Neural Network	**1.14**	**0.82**

TABLE II
RMSE and MAE for Algorithms below on likeness

Algorithm	RMSE	MAE
SVD	0.029	0.018
Basic Neural Network(without metadata)	0.023	0.10
Neural Collaborative Filtering	0.021	0.107
AutoEncoders based Neural Network	**0.019**	**0.098**
SVD + Neural Network	0.05	0.03

Based on the RMSE and MAE values obtained from our experiments, we can conclude that the SVD + NN approach had the best performance on the recommendation task, with RMSE and MAE values of 1.14 and 0.82, respectively. The AutoEncoders based Neural Network had the second-best performance, with RMSE and MAE values of 1.20 and 0.86, respectively. The Basic Neural Network (without metadata) had the third-best performance, with RMSE and MAE values of 1.23 and 0.93, respectively. The worst performing algorithm was the Neural Collaborative Filtering, with RMSE and MAE values of 1.8 and 1.3, respectively. However, it is worth noting that all the algorithms had relatively low RMSE and MAE values, indicating that they all performed reasonably well on the recommendation task.

Regarding the likeness prediction task, the AutoEncoders based Neural Network had the best performance, with RMSE and MAE values of 0.019 and 0.098, respectively. The Neural Collaborative Filtering had the second-best performance, with RMSE and MAE values of 0.021 and 0.107, respectively. The Basic Neural Network (without metadata) had the third-best performance, with RMSE and MAE values of 0.023 and 0.10, respectively. The SVD had the worst performance, with RMSE and MAE values of 0.029 and 0.018, respectively. However, it is worth noting that all the algorithms had relatively low RMSE and MAE values, indicating that they all performed reasonably well on the likeness prediction task.

TABLE III
RMSE and MAE for Artist Popularity model

Algorithm	RMSE	MAE
Auxiliary Output	0.102	0.082
Final Output	**0.015**	**0.0139**

Based on the results, we can see that the neural network model with auxiliary input and output achieved better performance than the model with only a final output layer. The RMSE and MAE values for the final output model were 0.015 and 0.0139 respectively, which were significantly lower than those of the auxiliary output model with RMSE and MAE values of 0.102 and 0.082 respectively. This indicates that considering artist familiarity as an input variable improved the model's ability to predict artist hotness or popularity accurately.

Furthermore, the results suggest that artist familiarity does have an impact on artist popularity. By including this input variable in the model, it provides more information to the model, which improves its ability to predict the target variable. This finding could be useful for music industry professionals who are interested in identifying the factors that contribute to an artist's popularity.

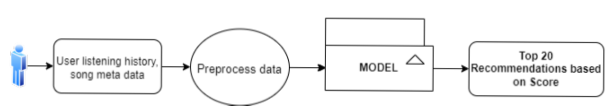

Fig. 10. Music Recommendation for Users

V. Conclusion

In conclusion, this project aimed to develop a music recommendation system using the Million Song Dataset, with a focus on the Taste profile dataset. The project involved various tasks such as data preprocessing, exploratory data analysis, feature engineering, and building and evaluating different recommendation algorithms. The goal was to enhance the recommendation quality by incorporating additional metadata such as artist and song information.

The project evaluated different recommendation algorithms using common evaluation metrics such as MSE and RMSE. The best performing model was found to be SVD + NN approach, which had an RMSE of 1.14 and an MAE of 0.82.

Lastly, the output variable was changed to likeness (song likeness = play count of the song listened by a user/ total song play counts of the user) to account for the lack of a rating concept in the dataset. This change allowed for a more accurate assessment of song popularity among users. Overall, this project demonstrated the potential of machine learning techniques in improving music recommendation systems, and the importance of incorporating additional metadata to enhance recommendation quality.

Additionally,One of the main experiments in the project focused on exploring the relationship between artist popularity and various song metadata, using lyrics data, word count, and NLP techniques. The results showed that incorporating artist familiarity had a positive impact on artist popularity prediction.

VI. Task Distribution

For this project, the team divided the tasks into different parts to effectively develop a music recommendation system using the Million Song Dataset. Firstly, a common preprocessing file was created for all the models to ensure consistency in the data processing. Exploratory data analysis (EDA) for metadata was done by Saiteja and on lyrics data was done by Sai Prasanna Kumar to gain insights and knowledge about the data.

Text processing was performed separately by each team member using different techniques such as tf-idf, Word2Vec, and FastText. The purpose of this was to test out different approaches and find the most effective one.

The team then proceeded to train the different models as suggested in the proposal. Baseline models such as SVD and base neural networks without metadata were trained initially by Neeharika. Next, the team worked on more complex models such as Neural Collaborative Filtering by Neeharika, Hybrid recommender system with autoencoders by Sai Prasanna Kumar, and SVD + NN by Saiteja.

In addition to the above models, a NCF model was trained to predict artist popularity based on song metadata and lyrics. This was done by Sai Prasanna Kumar.

Finally, all the models were run on two different dependent variables: play count and likeness of the song. The use of likeness as a dependent variable was necessary due to the lack of a concept of rating in the Million Song Dataset. Overall, the team's approach involved a collaborative effort to develop an effective music recommendation system using various models and techniques.

A. Challenges faced

During the development of the music recommendation system, the team faced several challenges. One of the main challenges was the inclusion of lyrics data. The team wanted to merge the lyrics data with the original play count user preference data and metadata. However, lyrics were missing for some songs listened to by users, which made it difficult to train the model. The team realized that if there are missing values in the data, the model would be biased towards non-null values. Therefore, they had to find a way to address this issue to ensure that the model was trained effectively and provided accurate recommendations.

Another challenge that the team faced was the implementation of similarity-based approaches. The dataset used in the project was huge, and it was not possible to create user-item or item-item based similarities. As a result, the team was unable to provide recommendations based on similarity. This was a significant challenge because similarity-based approaches are often effective in providing accurate recommendations. However, the team had to explore other options and came up with several recommendation algorithms that were effective in providing recommendations based on the available data.

References

[1] Bertin-Mahieux, T., Ellis, D. P., Whitman, B., & Lamere, P. (2011). The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), pp. 591-596.

[2] Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., et al. (2013). The MusiXmatch dataset for music text mining. In Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR 2013), pp. 557-562.

[3] A. A. Correya, R. Hennequin, and M. Arcos, "Large-Scale Cover Song Detection in Digital Music Libraries Using Metadata, Lyrics and Audio Features.," arXiv: Information Retrieval, Aug. 2018.

[4] D. Liang, R. G. Krishnan, M. D. Hoffman and T. Jebara, "Variational Autoencoders for Collaborative Filtering," in Proceedings of the 2018 World Wide Web Conference, Lyon, France, 2018, pp. 689-698, doi: 10.1145/3178876.3186150.

[5] J. Spijkervet and J.A. Burgoyne. "Contrastive Learning of Musical Representations." In Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR 2021), November 8-12, 2021, Virtual Event, Online, pp. 1-8.

[6] Strub, F., Gaudel, R., & Mary, J. (2016). Hybrid Recommender System based on Autoencoders. ArXiv. https://doi.org/10.1145/2988450.2988456.

[7] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. "The Million Song Dataset." In Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011), 591-596, 2011.