Music Listening Behavior: Forecasting Track Skipping

Ghina Al Shdaifat gha2009@nyu.edu
Ansh Jhaveri avj2013@nyu.edu
Kendall Brogle klb9313@nyu.edu

Milestone 1

1. Introduction

In an increasingly digitized entertainment industry, online music streaming services such as Spotify have seen exponential growth in usership. Changes in music development and distribution have led to revolutionary methods of music consumption. Users now enjoy vast libraries of music, causing a shift in how they interact with music. This research investigates the nuances of user interaction with Music Recommendation Systems (MRS) tasked with navigating extensive data to deliver personalized content aligning with the user's evolving tastes. Based on recent advances such as Deep Reinforced Learning (DLR), our work is focused on user engagement. The user feedback provided by track skipping, which serves as a crucial indicator of user satisfaction, can be used to advance MRS and improve user experience. By using historical data, this will analyze the likelihood of a user skipping a specific track during a listening session, how long into a track does this probability increase, and whether lyrics influence this probability.

2. Related Works

Sequential modeling of Sessions Using Recurrent Neural Networks For Skip Prediction[1]: This study was part of Spotify's sequential skip prediction challenge, predicting if users would skip tracks during the second half of a listening session. The study used data from Spotify, including 130 million listening sessions for training and 30 million for testing. It created a model that has a fixed vector representation of the listening session, which was used for an Encoder-Decoder style architecture typically used for sequential data like natural language processing. This study's strength includes its access to large amounts of data, a unique architecture, and its overall model performance with an average accuracy of 0.604. A drawback was the complex methodology, which can be computationally expensive. It may also risk over-fitting and an inability to perform well on unseen data, given the amount of training data it had access to. The fixed vector system may decrease the models ability to generalize to other data. It used the Mean Average Accuracy (MAA) to evaluate performance. The study used a weighted loss function, emphasizing earlier predictions. Music genre classification and music recommendation by using deep learning[2]: This study presents a new deep learning based approach of music genre classification and recommendation, called MusicRecNt. It classified music based on acoustic features with a neural network model. It used a convolution neural network (CNN) to extract features from music tracks and to classify and recommend songs. This was evaluated using the GTZAN data set which has 1,000 tracks spread across ten genres. Its advantages were that it outperformed previous methods with an average accuracy of 0.818 up to 0.976 when combined with an SVM classifier. The use of CNN and examination of window type size and overlap in time frequency analysis improved feature extraction. Negatively, the data set was small with only ten genres that may have shared overlap in qualities, making it at risk for overfitting with the training data.

3. Methodology

Recurrent Neural Networks (RNN), especially Long Short-Term (LSTM) are types of neural networks that determine sequence prediction problems. They will be used to model the progression of a song over time and predict at which point a user is likely to skip. These models are powerful, especially when using time-series data in which they can effectively capture the temporal dynamics within a track that lead to a skip. To determine whether lyrics influence the probability of skipping a song, we will be using Convolutional Neural Networks (CNN) for text classification. While CNNs are commonly associated with image processing, they are effective for NLP tasks and could be used to extract features from lyrics that are predictive of skips. By using NLP techniques along with the Spotify API, we will be able to transform the lyrics into numerical data, enabling us to create features that may contribute to skip probability as well as input features for our machine learning models. As the data has categorical and numerical features, Random Forest is useful, which uses a multitude of decision trees. It is useful for classification and regression, providing insight into feature importance about what factors are most predictive of a skip. Gradient Boosting is also beneficial as it often aims to provide better predictive accuracy than a Random Forest, yet can be prone to overfitting if not properly tuned. To evaluate the models, we will be using metrics like accuracy, precision, recall, F1 score, and the AUC/ROC curve, which will provide insight into how well the models are performing in terms of the balance between the relevance of the prediction (precision) and the ability to identify all actual skips (recall), combined into a single metric (F1 score). A combination of the models listed can help answer the research question. The application of the models to the extracted data and lyrical features will allow us to predict skip behavior, providing insights into user listening habits and the influence of different track characteristics. Evaluating the models will focus on the accuracy of the predictions and the time-specific nature of skip events, as well as the relevance and completeness of the predicted skips.

4. Data Set

Fig 1: skip1, skip2, skip3: shows the number of times users skipped tracks, looks like most tracks were not skipped not skipped: shows most tracks were not skipped Hist user behavior is shuffle: most users listen with shuffle off, but many do keep it on premium: most users have premium Context switch: most users do not switch context

Fig 2: Increase in activity in the morning, which goes on to peak in the afternoon

Lowest activity is in the early morning and late night

Fig 3: Positive correlation between danceability and valence, which makes sense as more danceable tracks are usually happier
Positive correlation between energy and loudness
Negative correlation between acousticness, and energy and loudness, which indicates that the more acoustic a song, the more quiet and less energetic it is
Fig 4: Non-premium users skip tracks slightly more than premium users
Catalog: tracks from catalog have the lowest skip rate
Personalized playlist: Similar to catalog, but slightly higher
Editorial playlist: Higher skip rates than personalized and catalog, indicating that users are more selective/exploratory
User collection: Higher skip rate, which may be due to familiarity with the music which may potentially contribute to a lower tolerance for less like songs
Radio: High skip rate, which makes sense because radio playlists can't really be controlled by the user
Charts: Highest skip rate, which may be due to a similar reason as user collection, which is that people are more familiar with the songs and hear them so often that they know whether or not they want to listen to it immediately
Fig 5: There is really no correlation here between musical features and skipping. There is a positive correlation between energy and skip 1 and negative correlations between danceability and skip 3, and acousticness with skip 1 and skip 2. Still, though, all of these are very slight and most likely are not relevant.
Fig 6: Duration: most tracks are about 200 seconds long with a tail approaching the longer end
US popularity estimate: Overall very high (high 90s)
Acousticness: most tracks have a low acousticness which means more electronic sound
Danceability: Relatively even distribution (slightly left skewed)
Energy: Left skewed, indicating that most tracks are high energy
Tempo: Relatively evenly distributed with peaks around 100-150, which is in line with modern pop music
valence: Also pretty evenly distributed, indicating broad variety of moods in tracks

5. Work Plan

Milestone 1 Contribution: Kendall: Motivation and Related Works, Ansh: Data Set, Ghina: Methodology and Work Plan We found our topic of interest and realized that music recommendation systems have been explored many times, which is why we decided to add prediction and forecasting. We split the work up (as shown above). Before the submission deadline of Milestone 2, will be meeting often to ensure the implementation of code is accurate and to work together in overcoming any difficulties. After the Milestone 2 submission, we will finalize our code and each begin writing the analysis corresponding to each model. After the Milestone 3 submission, each member will record themselves explaining their methodology. Ghina will combine the individual recordings into one video. We will each create slides that correspond to our part of the project (GoogleSlides). Finally, Kendall will upload everything on the GitHub repository.

Appendix

Figure 1:

In [58]:
```python
boolean_features_log_mini = [col for col in log_mini_df.columns if log_mini_df[col].dtype == 'bool']

plt.figure(figsize=(15, 10))
for i, feature in enumerate(boolean_features_log_mini, 1):
    plt.subplot(3, 3, i)
    log_mini_df[feature].value_counts().plot(kind='bar')
    plt.title(feature)
    plt.xticks(rotation=0)

plt.tight_layout()
plt.show()
```

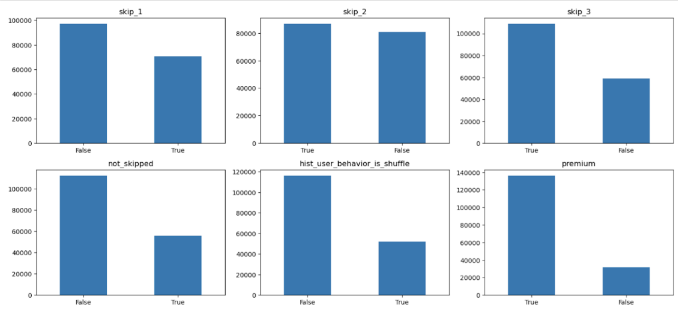


Figure 2:

In [59]:
```python
import seaborn as sns

correlation_matrix = tf_mini_df[numerical_features_tf_mini].corr()

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm')
plt.title('Correlation Heatmap for Numerical Features in tf_mini.csv')
plt.show()
```

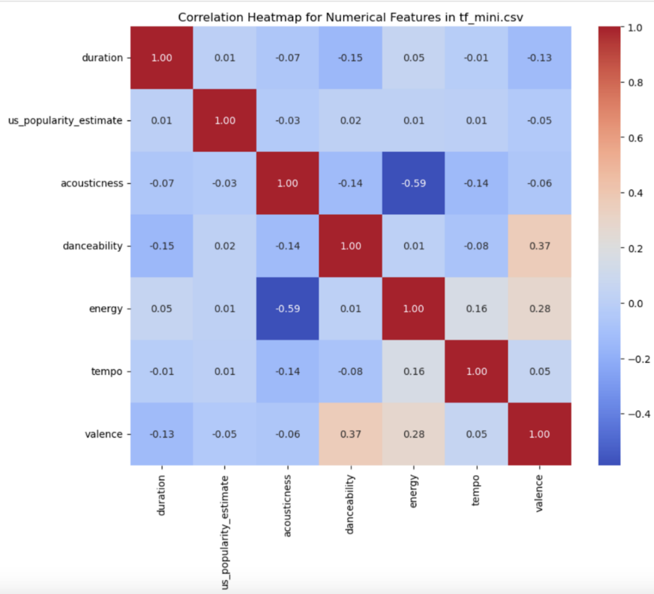


Figure 3:

```
In [60]: hourly_distribution = log_mini_df['hour_of_day'].value_counts().sort_index()

         plt.figure(figsize=(12, 6))
         hourly_distribution.plot(kind='bar')
         plt.title('Distribution of Track Plays Across Different Hours of the Day')
         plt.xlabel('Hour of Day')
         plt.ylabel('Number of Track Plays')
         plt.xticks(rotation=0)
         plt.show()
```

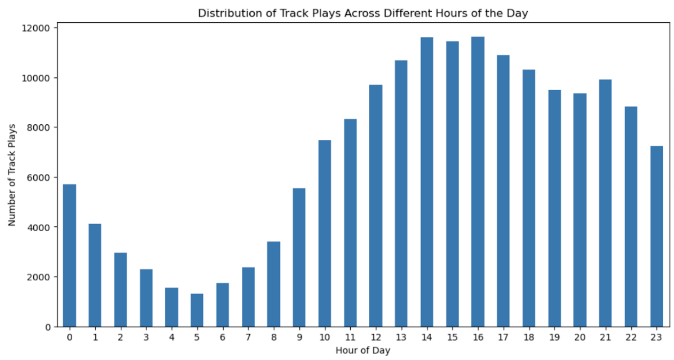


Figure 4:

```
In [61]: skip_columns = ['skip_1', 'skip_2', 'skip_3', 'not_skipped']
         premium_skipping_rates = log_mini_df.groupby('premium')[skip_columns].mean()

         context_skipping_rates = log_mini_df.groupby('context_type')[skip_columns].mean().sort_values(by='not_skipped', asce

         (premium_skipping_rates, context_skipping_rates)

Out[61]: (          skip_1    skip_2    skip_3  not_skipped
          premium
          False    0.449147  0.533761  0.652936     0.330254
          True     0.415037  0.513300  0.647314     0.333495,
                                skip_1    skip_2    skip_3  not_skipped
          context_type
          catalog              0.345059  0.456525  0.609309     0.371063
          personalized_playlist 0.331989  0.452285  0.632392     0.350470
          editorial_playlist   0.402490  0.508174  0.640131     0.341839
          user_collection      0.469109  0.546404  0.664022     0.316091
          radio                0.449282  0.553750  0.681196     0.303454
          charts               0.451454  0.555100  0.684965     0.300287)
```

Figure 5:

```
In [62]: merged_df = pd.merge(
             left=log_mini_df,
             right=tf_mini_df,
             how='left',
             left_on='track_id_clean',
             right_on='track_id'
         )

         merged_df.drop(columns=['track_id_clean', 'track_id'], inplace=True)

         musical_features = numerical_features_tf_mini
         skip_behaviors = ['skip_1', 'skip_2', 'skip_3', 'not_skipped']

         skip_correlation_matrix = merged_df[musical_features + skip_behaviors].corr().loc[musical_features, skip_behaviors]

         plt.figure(figsize=(10, 8))
         sns.heatmap(skip_correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm')
         plt.title('Correlation Heatmap Between Musical Features and Skipping Behavior')
         plt.show()
```

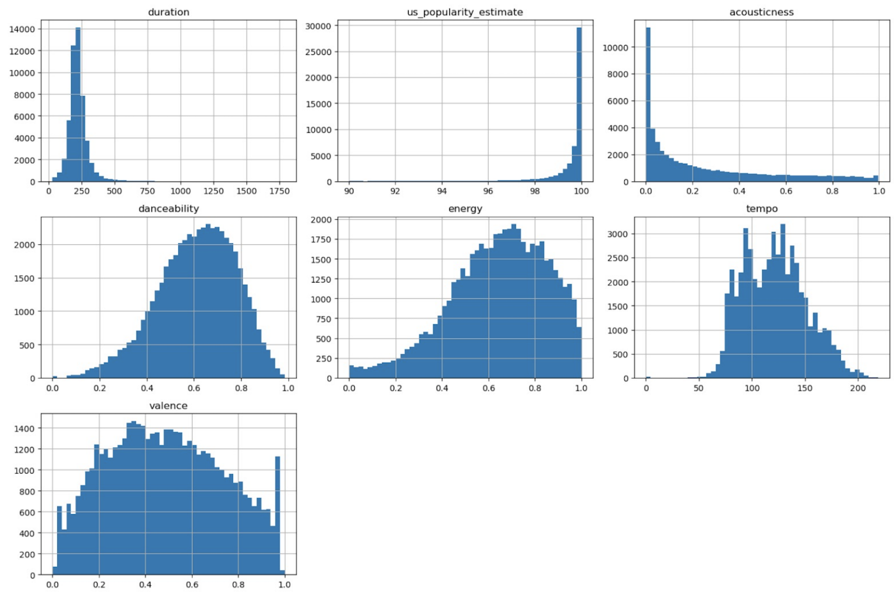
Figure 6:

```
In [57]: import matplotlib.pyplot as plt

         numerical_features_tf_mini = ['duration', 'us_popularity_estimate', 'acousticness', 'danceability', 'energy', 'tempo

         plt.figure(figsize=(15, 10))
         for i, feature in enumerate(numerical_features_tf_mini, 1):
             plt.subplot(3, 3, i)
             tf_mini_df[feature].hist(bins=50)
             plt.title(feature)

         plt.tight_layout()
         plt.show()
```



References

[1] Sainath Adapa. 2019. Sequential modeling of Sessions using Recurrent Neural Networks for Skip Prediction. arXiv preprint arXiv: 1904. 10273 (2019)
[2]Elbir, A. and Aydin, N. (2020), Music genre classification and music recommendation by using deep learning. Electron. Lett., 56: 627-629. https://doi.org/10.1049/el.2019.4202.