Groove Trends: Decoding Music Popularity

ACM Reference Format:
. 2023. Groove Trends: Decoding Music Popularity. *ACM Trans. Graph.* 37, 4, Article 111 (August 2023), 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Abstract

This research strikes a chord in the field of Music Information Retrieval by examining the factors influencing music popularity. Utilizing the Free Music Archive (FMA) dataset, which comprises over 100,000 songs across diverse genres, our study presents a symphony of data analysis and predictive modeling. Initially, data cleaning set the stage by filtering out discordant data, followed by an exploratory analysis that danced through variables like album and track favorites, listen counts, and genre impact. The crescendo of our study involved developing a custom 'popularity score', harmonizing listen and favorite counts with a double emphasis on listens, echoing the belief that listens play a stronger tune in indicating popularity. Our methodological ensemble included Linear Regression, Support Vector Machine (SVM), LightGBM, and Random Forests, with a focus on reducing the cacophony of overfitting and ensuring the models were music to the ears of accuracy. We conducted a comprehensive ensemble performance, tuning each model to the unique rhythms of our dataset. This approach allowed us to conduct a comparative analysis, identifying which algorithms best predicted the hits from the misses. The finale of our study revealed that Random Forest conducted the most harmonious predictions, orchestrating a balance between complexity and performance. This model struck the right note in capturing the subtle dynamics of music popularity, outperforming others in Root mean squared error (RMSE).

1 DATASET

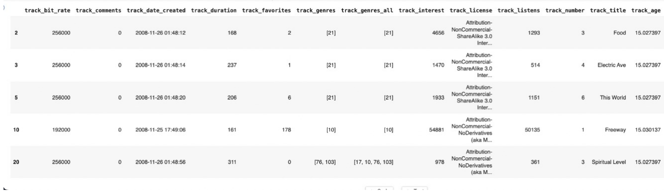
1.1 Introduction

We selected a custom-made dataset, the Free Music Archive (FMA), which contains artist and track metadata for 106,574 musical songs in over 150 genres. Since the dataset has been drawn from free music with fairly permissive licenses, it proves rich in data useful for complex and diverse Music Information Retrieval tasks. According to the researchers who compiled this dataset, the dataset provides "full-length and high-quality audio, pre-computed features, together with track- and user-level metadata, tags, and free-form text such as biographies." [1]

1.2 Data Cleaning

The initial dataset required some review and cleaning, since it was designed to reflect the complexities of music-related, real-world data. Therefore, in order to improve the clarity of the dataset before exploratory analysis, we first removed rows that contained over 70% of fields null. We chose 70% as a threshold because for most data fields, over 80% of the data had surpassed this threshold.

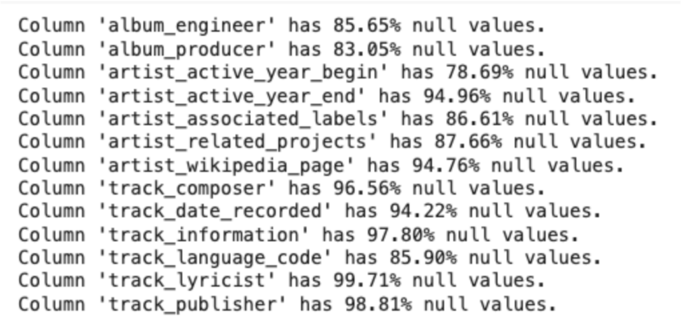
Fig. 1. Portion of Cleaned Dataset Head



	track_bit_rate	track_comments	track_date_created	track_duration	track_favorites	track_genres	track_genres_all	track_interest	track_license	track_listens	track_number	track_title	track_age
2	256000	0	2008-11-26 01:48:12	168	2	[21]	[21]	4656	Attribution-NonCommercial-ShareAlike 3.0 Inter...	1293	3	Food	15.027397
3	256000	0	2008-11-26 01:48:14	237	1	[21]	[21]	1470	Attribution-NonCommercial-ShareAlike 3.0 Inter...	514	4	Electric Ave	15.027397
5	256000	0	2008-11-26 01:48:20	206	6	[21]	[21]	1933	Attribution-NonCommercial-ShareAlike 3.0 Inter...	1151	6	This World	15.027397
10	192000	0	2008-11-25 17:49:06	161	178	[10]	[10]	54881	Attribution-NonCommercial-NoDerivatives (aka M...	50135	1	Freeway	15.030137
20	256000	0	2008-11-26 01:48:56	311	0	[76, 103]	[17, 10, 76, 103]	978	Attribution-NonCommercial-NoDerivatives (aka M...	361	3	Spiritual Level	15.027397

For the remaining rows containing less than 70% null values, we replaced NaN values with "Unknown". Aside from addressing null values, we also removed columns for "album tags" and "track tags" as these were largely empty and we felt that they simply created unnecessary noise. Additional, since date columns such as album release date and artist active beginning and ending dates were initially stored as text, we converted these to datetime data format so as to facilitate our analysis.

Fig. 2. Data Fields with more than 70% Null Values



```
Column 'album_engineer' has 85.65% null values.
Column 'album_producer' has 83.05% null values.
Column 'artist_active_year_begin' has 78.69% null values.
Column 'artist_active_year_end' has 94.96% null values.
Column 'artist_associated_labels' has 86.61% null values.
Column 'artist_related_projects' has 87.66% null values.
Column 'artist_wikipedia_page' has 94.76% null values.
Column 'track_composer' has 96.56% null values.
Column 'track_date_recorded' has 94.22% null values.
Column 'track_information' has 97.80% null values.
Column 'track_language_code' has 85.90% null values.
Column 'track_lyricist' has 99.71% null values.
Column 'track_publisher' has 98.81% null values.
```

Finally, we split our dataset into training, test, and validation sets. We decided to follow the recommendation of the researchers who compiled the dataset and split the data 80/10/10 between the three sets. We agreed given the complexity of the data that this split was optimal for any range of predictive tasks.

1.3 Exploratory Data Analysis

From the onset, we felt drawn to some kind of popularity prediction for albums, tracks, or both. As a measure for popularity, we used the data fields for album and track favorites and listens counts. We began by analyzing summary statistics for this data. We found that

[1] Defferrard, M., Benzi, K., Vandergheynst, P., & Bresson, X. (2017). Retrieved from https://arxiv.org/abs/1612.01840

Author's address:

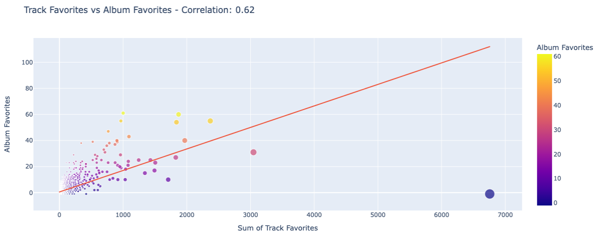
0730-0301/2023/8-ART111 $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

albums and tracks had means of 32,120 and 2,329 listens respectively and standard deviations of 147,853 and 8028.07 respectively. Album and Track favorites were much smaller in scale, with means of 1.29 and 3.18 respectively, as well as standard deviations of 3.13 and 13.51 respectively.

In order to identify an avenue of predictive analysis for album or track popularity within the dataset, we spent a considerable amount of time exploring the relationships of what we already believed were key variables. In particular, we initially examined popularity through the lens of the correlations and relationships between album and track favorites, album release date, track and album genre, and number of comments.

First, we determined the correlation between the count of album favorites and track favorites. A significant correlation between these variables could reveal the potential of track favorites as a predictor of album favorites or vise versa. We grouped the dataset by album and aggregated track and album favorite counts by taking the total and average respectively. Most albums had less than 20 favorites on average, whereas many individual tracks within the albums contained from several hundred to several thousand tracks. From our calculations we found that the count of a track's favorites do have a positive correlation of 0.62 with the associated album's favorite counts. We intuited that is common for a listener to like select tracks from an album but not the album as a whole; however, track favorites did seem to contribute to overall album favorites as well.

Fig. 3. Track and Album Favorites Correlation Plot



We continued with a closer look at the relationship between Track Age and Number of Listens, since we believed the longer a track has been available to the public, the more time people have had to listen to it. Surprisingly, we found a very small, negative correlation of -0.04 between track age and listens. Even so, we did not completely rule it out as a potential predictor variable.

We then grouped and sorted all tracks by the genre field, aggregating by number of listens, and determining the top 20 genres by the number of listens overall. From this we found that within the top 20 track genres in the dataset, Electronic, Soundtrack, Instrumental, and Experimental fall in the top four genres at both the album and track levels. At the track level, these genres contained listening counts of 63, 42,41, and 30 million respectively. At the album level, they had listen counts of 768, 567, 514, 390 million respectively. Plotting the top 20 genres showed us that there was in fact a significant difference in listen counts between these top 4 genres and the remaining 16. Given the considerable amount of spread, we felt that perhaps genre was worth considering as a popularity predictor in our future model.

Fig. 4. Track Age and Album Listens

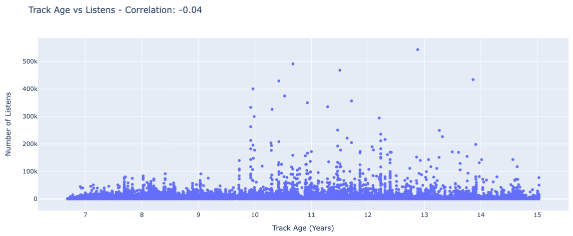
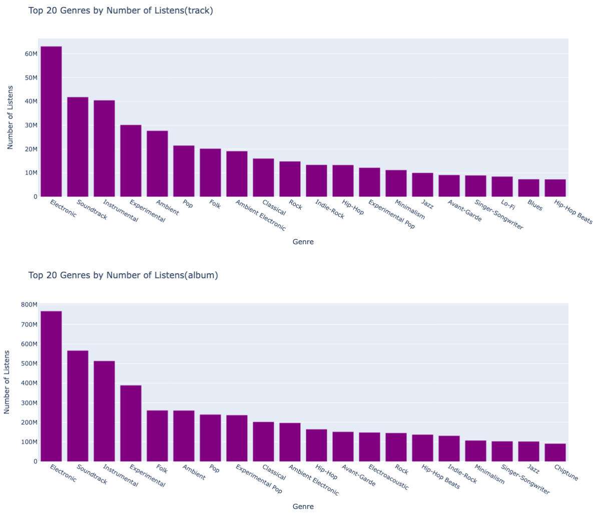


Fig. 5. Top 20 Track and Album Genres by Listen Counts

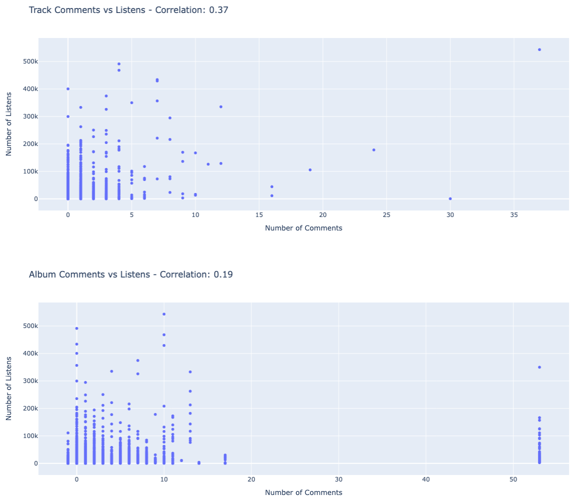


Our final consideration before model building was the relationship between count of track or album comments and listen counts. In this case, we found a correlation of only 0.19 and 0.37 between listen counts and track comments an album comments respectively. Given the relatively small size of this correlation, we were not sure if this would be valuable in prediction of track or album popularity.

2 PREDICTIVE TASK

After completing our initial data exploration, we decided to pursue our initial interest in both track and album popularity prediction separately. In order to consider both listen and favorite counts in our dual-measure of popularity, we calculated a custom "popularity score", adding these two values together and double-weighing track listens. We decided to give listens more weight in our calculation for both albums and tracks because we reasoned that any popular song, such as "ME!" from Taylor Swift's *Lover* album, could have a significant listen count, but a much smaller favorite count. We felt that listen count intuitively was a stronger indicator of interaction with tracks and albums and thus popularity overall.

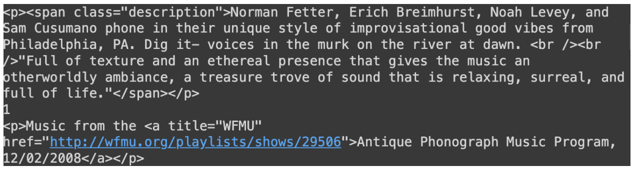
Fig. 6. Number of Comments v. Album and Track Listen Counts


Track Comments vs Listens - Correlation: 0.37

Album Comments vs Listens - Correlation: 0.19

For our initial, baseline model for track popularity score, we chose to use a linear regression model with predictors of genre, track age, track duration, track comments, track title, album favorites. For our album linear regression baseline, we used album comments, track favorites, album information, and various album type variables for form our album popularity predictions. Before building this model we had to one-hot-encode all top 20 genres and all top album types that we had earlier identified as well as a miscellaneous or "Other" grouping for the remaining genres and album types. In addition, we converted the text in the track titles and album information variables to token vectors using the Gensim *Word2Vec* method. Generating numerical representations of track titles allow for easier analysis and comparison of different tracks. This opened the door for dimensionality reduction within our model while still maintaining the semantic value of track titles. For album prediction, we also had to use BeautifulSoup to filter text from HTML in album information fields.

Fig. 7. Excerpt of Raw Album Info with HTML Syntax



```
<p><span class="description">Norman Fetter, Erich Breimhurst, Noah Levey, and
Sam Cusumano phone in their unique style of improvisational good vibes from
Philadelphia, PA. Dig it- voices in the murk on the river at dawn. <br /><br
/>"Full of texture and an ethereal presence that gives the music an
otherworldly ambiance, a treasure trove of sound that is relaxing, surreal, and
full of life."</span></p>
1
<p>Music from the <a title="WFMU"
href="http://wfmu.org/playlists/shows/29506">Antique Phonograph Music Program,
12/02/2008</a></p>
```

Based on our understanding of the dataset, as well as the assumptions required for such a linear regression model, we knew we would have to consider better alternatives for our predictions of popularity. In particular, our baseline model, as with most linear regression models, assumes a linear relationship between the predictor variables and popularity label. Considering the complexity of music information retrieval and the variety and size of our dataset, we opined that this assumption would not be realistic.

Therefore, to improve our baseline, we additionally considered methods such as Support Vector Machine (SVM), LightGBM, and Random Forests in our process of model selection and comparison. We assessed and compared the performance and validity of each model using the Root Mean Squared Error (RMSE). We selected RMSE over MSE for comparison due to the fact that it is more sensitive to outliers and also in the same scale and unit as our target.

3 MODEL AND RESULTS

3.1 Optimal Model Selection

After testing various model selection methods, we found that a Random Forest-generated model performed best in the prediction of both album and track popularity scores.

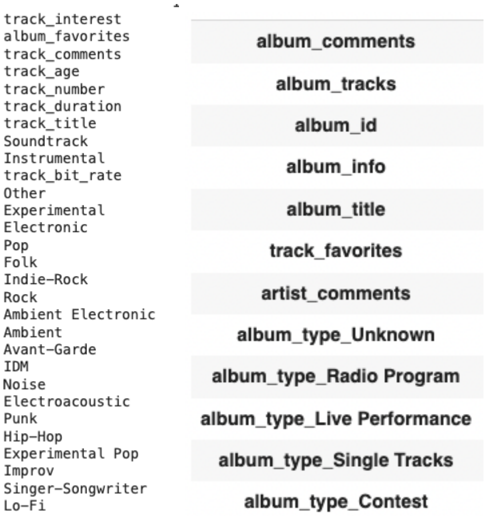
From our prior research and general knowledge, the Random Forest algorithm generates multiple random decision tree structures on the data. It uses the results of such decision trees, taking their average a prediction of the target. Overall, the goal of such a model is to fine-tune a set of hyperparameters, such as the number of decision trees in the forest and the conditions for each decision tree node split among others, until the combination of hyperparameters with the best predictive outcome has been reached.

To run this algorithm on our training set, we first defined ranges for our hyperparameter values. For example, we used the collection of integers from 200 to 1000 as potential values for the number of trees in our forest. Applying such ranges allowed the algorithm to explore a wider array of possible parameter combinations. We then built a "hyperparameter grid", combining these values into many different combinations. This grid served as a blueprint for the algorithm's hyperparameter search, with each combination representing a different configuration of our model for popularity prediction.

To evaluate how well each configuration performed, we used a technique called "cross-validation". This involves splitting the music dataset into multiple subsets, training each model on some of these subsets, and evaluating its performance on others. The benefit of this method was that it helped us ensure that the model's performance was not overly dependent on one specific data split and thus could be generalized to other data splits. We used a specific type of cross-validation called "RepeatedKFold," which performed cross-validation multiple times on a random number, *k*, data splits.

Ultimately, we selected the Random Forest Model for several reasons. First, it had the lowest RMSE of all models we considered. Additionally, we felt that it most effectively captured non-linearity in our dataset. Furthermore, while some might argue that this method of brute-force comparison of many hyperparameter combinations is inefficient, we found it beneficial for our purposes. The inherent complexity and high-dimensionality of our music-based popularity prediction tasks made finding an accurate shortcut method nearly impossible. Finally, this model attributed importance scores to each of the predictor variables, which provided us with some insight as to the relative weight of each in determining album and track popularity.

Fig. 8. Features Used in Track and Album Popularity Prediction Models



```
track_interest
album_favorites
track_comments
track_age
track_number
track_duration
track_title
Soundtrack
Instrumental
track_bit_rate
Other
Experimental
Electronic
Pop
Folk
Indie-Rock
Rock
Ambient Electronic
Ambient
Avant-Garde
IDM
Noise
Electroacoustic
Punk
Hip-Hop
Experimental Pop
Improv
Singer-Songwriter
Lo-Fi
```

album_comments
album_tracks
album_id
album_info
album_title
track_favorites
artist_comments
album_type_Unknown
album_type_Radio Program
album_type_Live Performance
album_type_Single Tracks
album_type_Contest

3.2 Other Models Considered

For both track and album popularity predictions, we began with our baseline Linear Regression model containing one-hot-encoded genre values and vectorized text fields such as track title and album info. For each label, track and album popularity, we also included associated album or track favorites respectively. In both cases, this model performed very poorly, which believed was largely due to its inability of to capture non-linear patterns in our diverse dataset, as well as its high sensitivity to outliers.

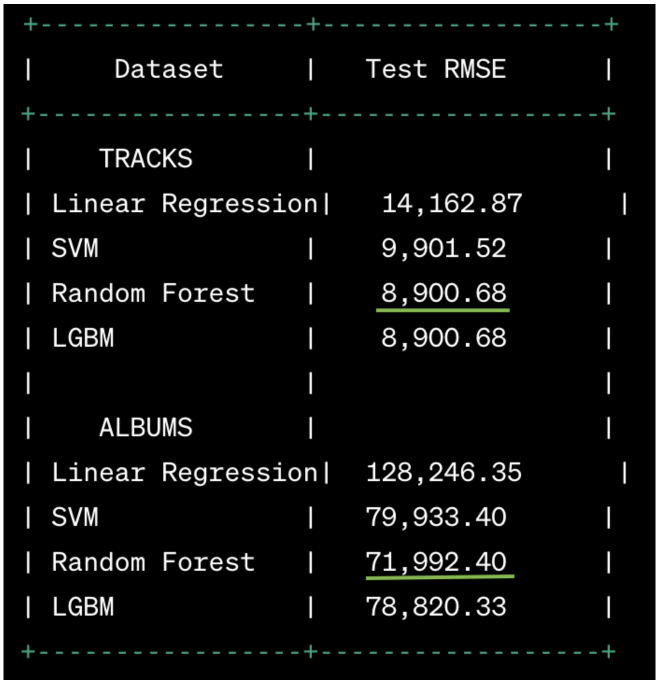
We subsequently tested the performance of a model that we generated from Support Vector Machine (SVM) methods. Like, Random Forest Regression, SVM evaluated our training data with the selection of hyperparameters. SVM, however, attempts to find the optimal division or grouping of the data so as to fit data points in a way that minimizes the prediction error of our target variable. Unfortunately, while SVM had improved our RMSE from the linear baseline models for album and track popularity, we still felt that we had room for additional improvement. While the algorithmic aspect of this model certainly contributed to its strength, it paled in comparison to the Random Forest algorithm. In particular, we found that SVM required careful selection of hyperparameters such as regularization values. This contrasts with the Random Forest method, which investigates all combinations given within defined ranges. In addition, the SVM method was extremely computationally intensive, especially with this large dataset. Of all model generators we tested, SVM took the longest to compute by a longshot. Additionally, we determined that the SVM model likely did not fit as well with our data as it is less robust with high-dimensional or noisy datasets.

As a good-faith measure, after using the Random Forest algorithm, we used a LightGBM Regressor to see if we could obtain a better error value. Similar to Random Forest Regression, LightGBM bases model selection on a host of decision trees; however unlike Random Forest, which builds decision trees during model training and ultimately combines their predictions through averaging, LightGBM creates these trees sequentially. Each subsequent decision tree serves to "boost" or correct the errors of the previous one. LightGBM did not end up changing the RMSE for track popularity prediction from the optimal Random Forest model. Nevertheless, in spite of some similarity to Random Forest, the LightGBM model increased our error for album popularity prediction by a 6,000. We found the most probable explanation for this increase was the susceptibility of LightGBM to overfitting when hyperparameters are not tuned adequately.

3.3 Results

After researching and testing out all models and comparing RMSE's for track and album popularity-score predictions, we landed on our optimal Random Forest model in both cases. In the end, the main deciding factor between LightGBM and Random Forests was the higher album prediction RMSE of the former.

Fig. 9. Model RMSEs For Both Tasks



```
+----------------+-----------------+
|    Dataset     |   Test RMSE     |
+----------------+-----------------+
|   TRACKS       |                 |
| Linear Regression|   14,162.87     |
| SVM            |   9,901.52      |
| Random Forest  |   8,900.68      |
| LGBM           |   8,900.68      |
|                |                 |
|   ALBUMS       |                 |
| Linear Regression|  128,246.35     |
| SVM            |  79,933.40      |
| Random Forest  |  71,992.40      |
| LGBM           |  78,820.33      |
+----------------+-----------------+
```

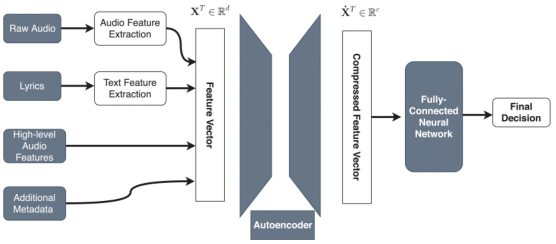
For all the models we tested, apart from considering the resulting RMSE for track and album-level predictions, we also took into account how the inherent qualities of each model fit with the high-dimensional, complex structure of our dataset. Even though the Random Forest Model arguably seems quite intensive, it was not nearly as computationally intensive and inefficient as the SVM and LightGBM models. In addition, we considered some, reasonable level of brute-force analysis necessary given the structure of our dataset.

4 LITERATURE

In their article FMA: A Dataset For Music Analysis, the creators of this dataset explain that they intended to create an easily accessible dataset rich in artist and album metadata for Music Information Retrieval (MIR) tasks. They developed their dataset for such tasks as "Advanced Genre Recognition" and "Music Classification" by retrieving calling from an API a collection of song data curated by artists who have chosen to make their music readily available without legal restriction. One issue with this dataset is that it would not be useful in predicting major hits since it is more centered around “experimental, electronic, and rock music” and does not “does not contain mainstream music and few commercially successful artists."

We reviewed other literature and prediction methods within this realm and found that many also involved dimension reduction. Our favorite top-hit prediction model described in a 2020 article in fact used a custom dataset, SpotGenTrack (SPD) [2], as well as an Autoencoder to compress a vector extracted song features, including audio, lyrics, and metadata, and cross-validation to identify optimal hyperparameters. The results of their predictions led to MSE’s as low as 0.0118, which is remarkably more accurate than our optimal model.

Fig. 10. Example Outline of Prediction Process for Similar Research


$\mathbf{X}^T \in \mathbb{R}^d$
$\hat{\mathbf{X}}^T \in \mathbb{R}^c$
Raw Audio
Audio Feature Extraction
Lyrics
Text Feature Extraction
High-level Audio Features
Additional Metadata
Feature Vector
Autoencoder
Compressed Feature Vector
Fully-Connected Neural Network
Final Decision

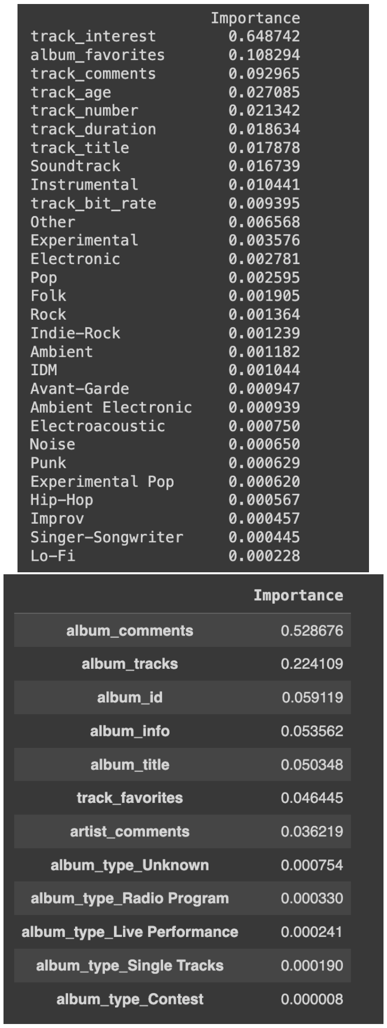
5 CONCLUSION

In sum, the Random Forest models had the lowest RMSE for track and album popularity prediction out of all the models that we tested. Although the RMSE of our final model was still quite high, 8,900 for tracks and 72,000, it was a vast improvement from our baseline model, and we believe the scale is largely due to the high dimensionality of the dataset.

While testing different models, the primary issues that we faced were with handling of the different data types, such as text, as well as hyperparameter tuning in the LightGBM and Random Forest Models. We used methods such as Word2Vec and BeautifulSoup to handle text data types. In terms of hyperparameter tuning, we found that the Random Forest model performed best of all algorithms we tried because it tested a diverse range of parameter combinations during the training phase. It proved much more difficult to choose correct hyperparameters using LightGBM or SVM.

We based some of the choices about the variable selection on logic and statistical analysis. From our optimal models for each task, we found that for track predictions the most significant predictors were "track_interest" and "album_favorites", whereas the most significant predictors for albums were "album_comments" and "album_tracks". In both models, the least significant predictors were amongst those that were one-hot-encoded.

Fig. 11. Track and Album Importance Scores



```
                    Importance
track_interest        0.648742
album_favorites       0.108294
track_comments        0.092965
track_age             0.027085
track_number          0.021342
track_duration        0.018634
track_title           0.017878
Soundtrack            0.016739
Instrumental          0.010441
track_bit_rate        0.009395
Other                 0.006568
Experimental          0.003576
Electronic            0.002781
Pop                   0.002595
Folk                  0.001905
Rock                  0.001364
Indie-Rock            0.001239
Ambient               0.001182
IDM                   0.001044
Avant-Garde           0.000947
Ambient Electronic    0.000939
Electroacoustic       0.000750
Noise                 0.000650
Punk                  0.000629
Experimental Pop      0.000620
Hip-Hop               0.000567
Improv                0.000457
Singer-Songwriter     0.000445
Lo-Fi                 0.000228
```

	Importance
album_comments	0.528676
album_tracks	0.224109
album_id	0.059119
album_info	0.053562
album_title	0.050348
track_favorites	0.046445
artist_comments	0.036219
album_type_Unknown	0.000754
album_type_Radio Program	0.000330
album_type_Live Performance	0.000241
album_type_Single Tracks	0.000190
album_type_Contest	0.000008

We found it interesting and logical that album favorites count was a significant predictor for track popularity, whereas track favorites did not significantly predict album popularity. It made sense that favorites on an entire album would correspond to track favorites within that album as well. The album comments also

Above all, we currently believe our research adds a new melody to the understanding of music popularity, suggesting that a blend of user engagement metrics, when tuned correctly, can predict a song’s chart-topping potential. This work not only amplifies the discourse in Music Information Retrieval but also provides practical insights for artists and producers, tuning their creations to the audience’s preferences. Future research may further explore the lyrical and acoustic features, adding more instruments to this analytical orchestra, and enhancing the predictive symphony of music popularity.

[2] Martín-Gutiérrez, D., Belmonte-Hernández, A., & Hernández Peñaloza, G. (n.d.). Retrieved from https://ieeexplore.ieee.org/abstract/document/9007339