Spotify Music Popularity Analysis *ISyE 6414: Regression*

Team Members:
Abhinav Arun: aarun60@gatech.edu
Ashish Dhiman: adhiman9@gatech.edu
Anshit Verma: averma373@gatech.edu
Kien Tran: ktran332@gatech.edu
Saksham Arora: sarora320@gatech.edu

1 Abstract

In this study, we explored the complex features associated with music and their impact on the popularity of a song on Spotify. By building a linear regression model, we identified key factors that contribute to the traction of a song, including the artist and other songs in the same album. We also found that musical attributes such as loudness and tempo play a significant role in explaining a song's popularity, given that non-musical attributes remain constant. Our findings indicate that the influence of these factors is stronger for newly-released songs and tends to plateau after a few years. Overall, this research provides valuable insights for musicians and industry professionals looking to understand and predict the popularity of their songs on Spotify.

2 Introduction

Music is a universal language that transcends boundaries and connects people from all walks of life. It has been an integral part of human civilization for centuries, with the music industry now valued at over $25 billion. In this study, we sought to understand the dynamics of music popularity and the various factors driving it. We explored complex features associated with music and external factors that contribute to a song's traction. Our goal was to build a predictive model that would allow us to not only forecast the popularity of a song but also identify specific factors that increase its propensity for popularity.

To achieve this, we followed a methodology pipeline that involved data collection, explanatory data analysis, feature engineering, modeling, and residual analysis. We used Spotify's API to

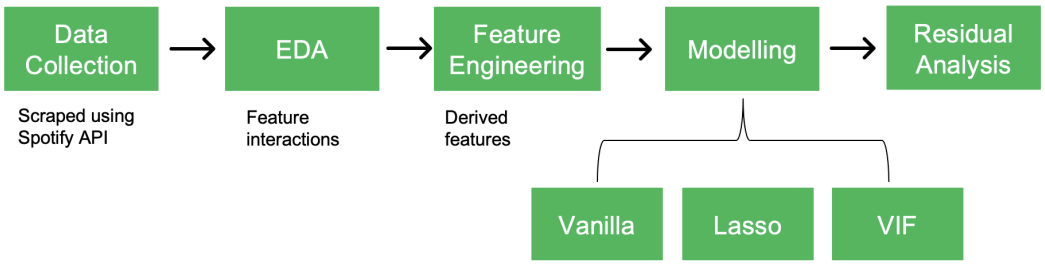


Figure 1: Methodology Pipeline

collect data on popular songs and artists and their attributes. We then conducted explanatory data analysis to identify key trends and patterns in the data. The initial analyses informed our next step - feature engineering. In this step, we collected additional data and created new meaningful variables. With the total of all features, we began the modeling phase with a baseline model, then performed feature selection using Lasso regression and variance inflation factor (VIF). Finally, we conducted residual analysis to assess the performance of our model and identify any potential issues or areas for improvement.

3 Data Collection

We have used Spotify API to fetch data about artists, albums, and tracks. We first started out by finding out the top 500 artists of all time[1]. Then we collected data from Spotify about their top 10 streamed tracks. The collected data includes the following fields:

- **Dependent variable:**
  - Popularity: Index defined by Spotify based on the total number of plays and their recency.
- **Independent variables:**
  - Audio Features: Danceability, Acousticness, Energy, Instrumentalness, Liveness, Loudness, Speechiness, Tempo, Time signature.
  - Logistical features: Name, Album, Artist, Length.

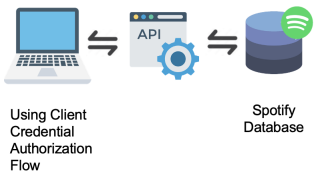


Figure 2: Data Collection Framework

Additionally, as a part of the feature engineering step, we also collected similar information regarding the other songs present in the albums of the tracks under consideration.

4 Exploratory Data Analysis (EDA)

As described in the introduction above, we want to build a regression model to predict the Popularity of a song (or track) basis it's intrinsic musical characteristics as well as other factors affecting popularity. The first step towards this goal is to understand the data scraped off Spotify API comprehensively, and then use it for our prediction model. As part of EDA, we have looked at a number of data characteristics such as:

4.1 Distribution of target variable (i.e. Popularity)

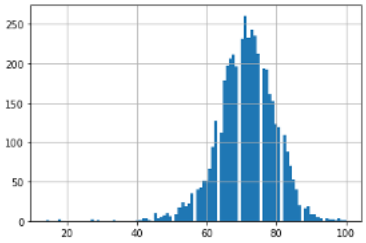


Figure 3: Histogram of Target Variable

As evident from the graph above, the popularity of the song follows an approximate normal distribution with a mean of around 71. The percentiles of the variable are given as follows:

percentile	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
popularity	62.0	65.0	68.0	70.0	72.0	74.0	76.0	78.0	81.0	100.0

Table 1: Percentiles for Track Popularity

4.2 Joint Distribution of target with predictors

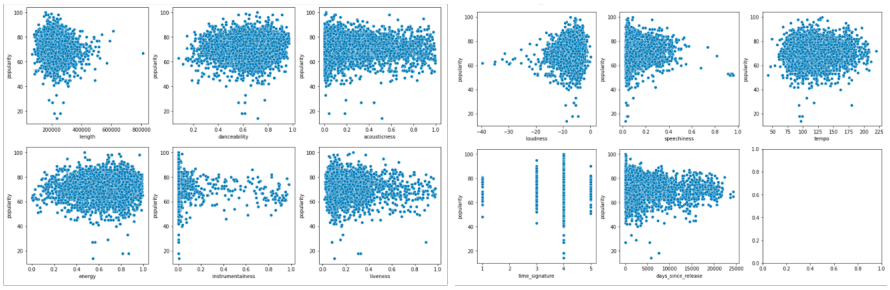


Figure 4: Scatter plot of target vs predictors

As seen from the above scatter plots, we see minimal linear relationship between the target and predictor variables.

5 Feature Engineering

As evident from the EDA section above, the original features scraped from the Spotify API lack significant predictive power against the target variable. Hence to mitigate this we have added our set of predictor variables, by engineering new predictors capturing intuitive relationships with the target variable.

5.1 Popularity of Album and Artist

It is only intuitive to hypothesize that the popularity of a song is hugely dependent on the popularity of the artist. In other words, if a popular artist releases a song, chances are that the new song will be popular too, while a song released by a relatively unknown artist will find it much harder to gain traction. Similar behaviour is also exhibited by albums.

Hence we have engineered features to create this exact same intuition, by creating the average popularity of the artist and the album excluding the original song itself. The scatter plots of these derived features against the target are given below, and evidently they exhibit a significant correlation with the target.

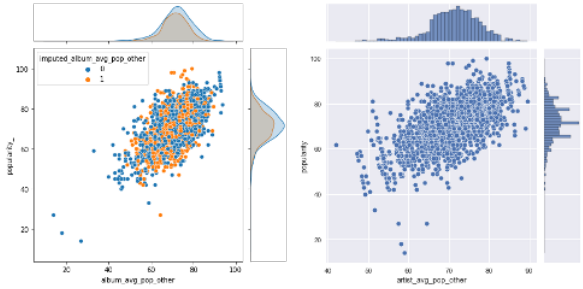


Figure 5: Average Album Popularity(left) and Average Artist Popularity(right)

5.2 Spatial Analysis of Artists

Some artists are more likely to produce music of a similar kind than others, for ex: Drake and PSY vs Drake and Justin Beiner. We hoped to create predictors, which divide artists into groups explaining these similarities and dissimilarities. To validate this hypothesis we can create scatter plots of various artists.

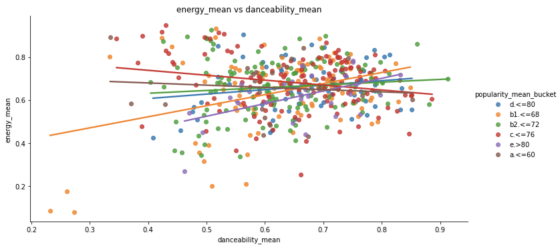
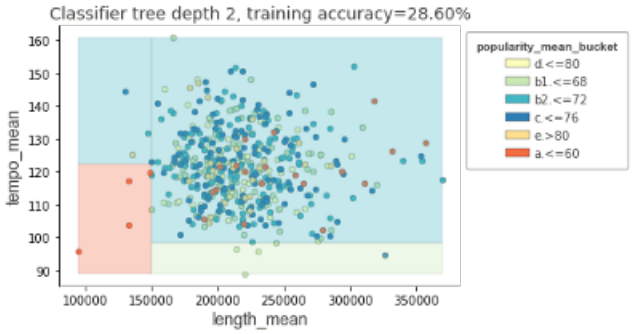


Figure 6: Scatter plot of Artist features

To create these groupings we have followed the following steps:

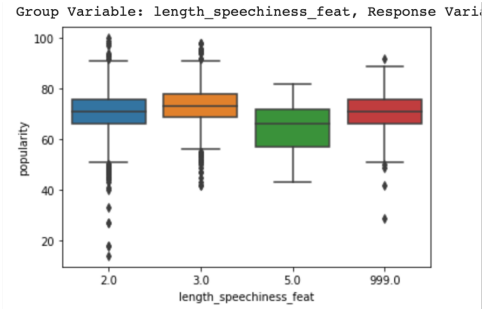
1. Aggregate the scraped data from the level of tracks to artists with average, i.e. calculate average of all features including popularity.
2. Create y_{bucket} after discretizing avg artist popularity into suitable buckets.
3. For all pairs of combinations in predictors at the level of artist, fit a depth 2 decision tree:
4. From the above fit decision tree generate predictions for artists. These prediction serve as the groupings.




```
|--- feature_0 <= 149865.46
|    |--- feature_1 <= 122.45
|    |    |--- class: 5
|    |--- feature_1 >  122.45
|    |    |--- class: 2
|--- feature_0 >  149865.46
|    |--- feature_1 <= 98.46
|    |    |--- class: 1
|    |--- feature_1 >  98.46
|    |    |--- class: 2
```

Figure 7: Decision Tree for a combination

5. After generating groupings for all possible predictors, choose the ones with highest drop in entropy.
6. Validate the level of track popularity with Tukey Test.



Multiple Comparison of Means - Tukey HSD, FWER=0.01

group1	group2	meandiff	p-adj	lower	upper	reject
2.0	3.0	2.342	0.001	1.4647	3.2193	True
2.0	5.0	-7.2288	0.001	-10.0771	-4.3806	True
2.0	999.0	-0.2646	0.9	-1.7836	1.2544	False
3.0	5.0	-9.5709	0.001	-12.4813	-6.6604	True
3.0	999.0	-2.6066	0.001	-4.2391	-0.974	True
5.0	999.0	6.9643	0.001	3.8007	10.1279	True

Figure 8: Tukey Test for a combination

6 Regression Modelling

After forming new variables to increase the predictive power of the model, we start by building a vanilla linear regression model. We consider all the variables and assess the performance by considering the 'Coefficient of Determination, R^2'.

	coef	std err	t	P>|t|	[0.025	0.975]
const	20.1692	5.547	3.636	0.000	9.236	31.102
length	-6.4982	4.848	-1.341	0.181	-16.053	3.057
danceability	0.0792	2.395	0.033	0.974	-4.642	4.800
acousticness	-0.7187	2.850	-0.252	0.801	-6.336	4.898
energy	-3.7760	3.047	-1.239	0.217	-9.782	2.230
instrumentalness	12.6152	7.998	1.577	0.116	-3.149	28.379
liveness	1.5767	2.517	0.627	0.532	-3.384	6.537
loudness	11.7989	6.147	1.920	0.056	-0.317	23.915
speechiness	2.3517	4.026	0.584	0.560	-5.584	10.287
tempo	-7.1755	2.367	-3.032	0.003	-11.841	-2.510
days_since	2.8295	1.837	1.541	0.125	-0.790	6.450
popularity_meta	17.1241	4.136	4.140	0.000	8.972	25.276
followers	7.4146	12.404	0.598	0.551	-17.034	31.863
popularity_count	3.7524	5.612	0.669	0.504	-7.309	14.814
popularity_count_album	-1.5721	3.144	-0.500	0.618	-7.770	4.626
artist_avg_pop_other	-6.0799	7.211	-0.843	0.400	-20.294	8.134
album_avg_pop_other	30.5589	6.758	4.522	0.000	17.238	43.880
imputed_album_avg_pop_other	1.6590	1.588	1.045	0.297	-1.470	4.788

OLS Regression Results

Dep. Variable:	popularity_	**R-squared:**	0.685
Model:	OLS	**Adj. R-squared:**	0.640
Method:	Least Squares	**F-statistic:**	15.11
Date:	Sun, 27 Nov 2022	**Prob (F-statistic):**	1.07e-38
Time:	18:37:09	**Log-Likelihood:**	-768.27
No. Observations:	247	**AIC:**	1601.
Df Residuals:	215	**BIC:**	1713.
Df Model:	31		
Covariance Type:	nonrobust		

Figure 9: Vanilla Linear Regression Model

As it can be seen above that the vanilla model is only able to explain 68.5% variability in our data. Another thing to observe is that many of the features are not statistically significant. Reducing the number of features can help us improve our model performance. We will be employing Lasso Regression to reduce the number of features. After obtaining significant features, we will use VIF to determine multicollinearity between variables.

6.1 Lasso Regression

As described above, in order to narrow down our model to features with significant predictive power, we used **LASSO Regression** as a feature selection technique. This also helped us deal with the multicollinearity problem which would be more evident from the VIF analysis done later.

Also , it is always good to have a lesser number of independent variables if the same amount of variability could be explained as it is sometimes expensive to collect data for some features. Thus , We iterated over different values of regularization parameter α and based on the values of the Sum of squared errors, R squared and Adjusted-R squared, we decided to go ahead with the value of 0.1 for the penalty parameter.

The same could be corroborated by the plots below.

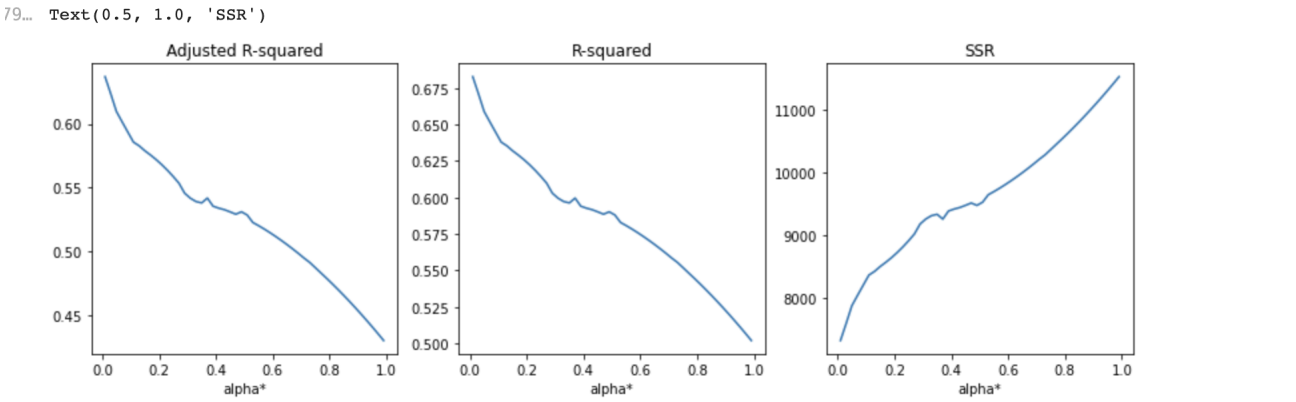


Figure 10: R-squared, Adjusted R-squared, and SSR for different values of regularization parameter

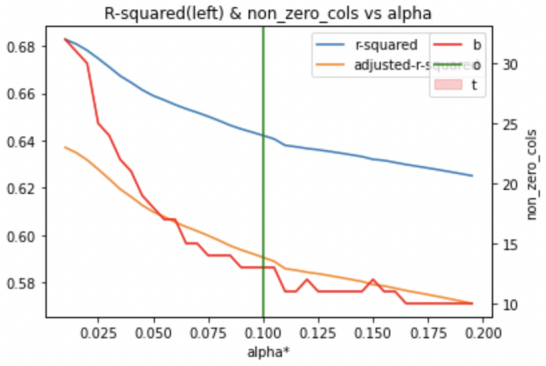


Figure 11: Finding the optimal value of Regularization parameter

The columns selected can be found in Figure 18 in Appendix.

OLS Regression Results

Dep. Variable:	popularity_	**R-squared:**	0.642
Model:	OLS	**Adj. R-squared:**	0.591
Method:	Least Squares	**F-statistic:**	12.44
Date:	Sun, 27 Nov 2022	**Prob (F-statistic):**	4.32e-33
Time:	18:50:46	**Log-Likelihood:**	-784.17
No. Observations:	247	**AIC:**	1632.
Df Residuals:	215	**BIC:**	1745.
Df Model:	31		
Covariance Type:	nonrobust		

Figure 12: Summary Output of the LASSO Regression Model

7 VIF

VIF aka **Variance Inflation Factor** is a measure of the amount of multicollinearity in regression analysis. If a given variable can be expressed a linear combination of other variables in the model, then that variable would have a very high VIF and should not be included in the model as it will cause the standard error of the coefficient estimates to be very large and we'll get very vague values of variable coefficients.

The plot below show the utility of applying a LASSO Regression Technique as several variables with high VIF is not selected by our regression model, and thus we get a set of features that are mostly uncorrelated.

	VIF	variable
0	272.969828	const
4	2.006594	album_avg_pop_other
3	1.448989	popularity_meta
1	1.349130	loudness
5	1.213831	imputed_album_avg_pop_other
2	1.089849	tempo
6	1.071862	days_since_log

Figure 13: VIF after fitting LASSO regression.

Thus, it is clearly evident from the above plot, that it is imperative to check for VIFs before proceeding with a given model inorder to avoid multicollinearity.

8 Final Model

After zeroing in on the right mix of features using Lasso and addressing multicollinearity, we built our final model and got the following results:

OLS Regression Results

Dep. Variable:	popularity_	**R-squared:**	0.662
Model:	OLS	**Adj. R-squared:**	0.649
Method:	Least Squares	**F-statistic:**	51.58
Date:	Sun, 27 Nov 2022	**Prob (F-statistic):**	5.98e-51
Time:	23:29:42	**Log-Likelihood:**	-777.11
No. Observations:	247	**AIC:**	1574.
Df Residuals:	237	**BIC:**	1609.
Df Model:	9		
Covariance Type:	nonrobust		

→

	coef	std err	t	P>|t|	[0.025	0.975]
const	20.8900	6.829	3.059	0.002	7.437	34.343
loudness	0.3515	0.173	2.035	0.043	0.011	0.692
tempo	-0.0350	0.013	-2.717	0.007	-0.060	-0.010
imputed_album_avg_pop_other	2.8240	0.776	3.637	0.000	1.295	4.354
danceability_speechiness_feat_3.0	-1.1408	0.820	-1.391	0.166	-2.756	0.475
length_speechiness_feat_999.0	4.1716	2.095	1.991	0.048	0.044	8.299
instrumentalness_time_signature_feat_2.0	8.4704	5.801	1.460	0.146	-2.958	19.899
instrumentalness_time_signature_feat_999.0	4.1716	2.095	1.991	0.048	0.044	8.299
length_tempo_feat_2.0	3.4963	2.340	1.494	0.137	-1.115	8.107
length_tempo_feat_999.0	4.1716	2.095	1.991	0.048	0.044	8.299
days_since_log	0.7040	0.163	4.324	0.000	0.383	1.025
popularity_meta_album_avg_pop_other	0.0076	0.001	14.838	0.000	0.007	0.009

Figure 14: Final Regression Model

8.1 Interpretation

As you can see R-squared is doing fairly well with a value of 0.662. Even after dropping multiple features, it is still comparable to the vanilla model of with R-sqaure of 0.68. Moreover, we see the model's robustness as almost all the variables come out to be significant under the 95% confidence interval. Additionally, since song features are intrinsic to a song, we can potentially draw causal conclusions. We can gauge the following insights:

- Intuitively enough, we notice that Artist and Album Popularity have a significant relation with song popularity
- Louder songs tend to become more popular
- Tempo has a slight negative relation with song popularity

8.2 Model Diagnostics

For checking the Linearity Assumption, we plotted residuals against the independent variables . We infer from the below plots that the linearity assumption holds for the most part. We might see that days_since_log plot seems odd. However, this is because we have data such that we had very few very recent songs, and most of the songs are older than a week. Thus data may look disjoint, errors are distributed equally above and below the zero line.

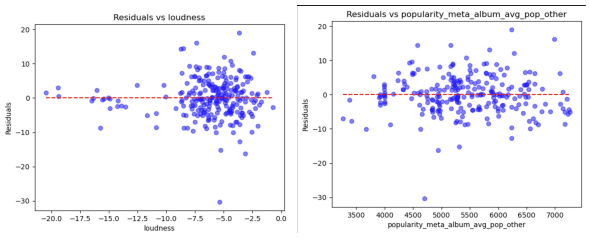


Figure 15: Residuals vs imputed_album_avg_pop_other and instrumentalness_time_signature_feat_2.0

Next, we checked Normality and Constant Variance assumptions, both of which seem to be inline. The histogram looks normally distritbuted and in the QQ Plot points line up along the

ideal line fairly well. The residuals are also randomly distributed about the 0 line when plotted against fitted values.

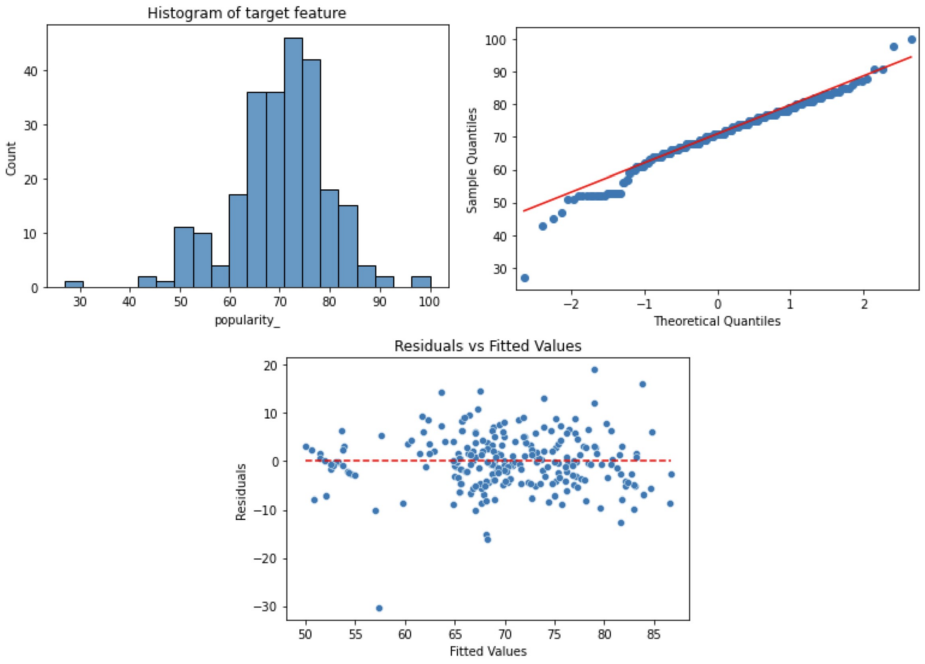


Figure 16: Normality Assumption and Constant Variance Check

9 Conclusion

Given the above model outputs, we can conclude that the popularity of a song is heavily influenced by the artist, and to be more specific, the popularity of other songs in the same album. The influence is stronger for the recently-released songs (R-squared up to 0.65). After 3 years since release day, the influence seems to plateau. Among musical attributes, 'loudness', and 'tempo' are the two most important factors in explaining a song's popularity, given that non-musical attributes like album popularity and release date remain fixed. Not all song features, even though they seem critical to a song's success, contribute towards it's popularity metric. Our final model was able to explain 66.2% variability in the data.

10 References

[1] J. S. Gulmatico, J. A. B. Susa, M. A. F. Malbog, A. Acoba, M. D. Nipas and J. N. Mindoro, "SpotiPred: A Machine Learning Approach Prediction of Spotify Music Popularity by Audio Features," 2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T), 2022, pp. 1-5, doi: 10.1109/ICPC2T53885.2022.9776765.

[2] "Most Streamed Artists on Spotify (Daily Update)." ChartMasters, 10 Oct. 2022, https://chartmasters.org/most-streamed-artists-ever-on-spotify/.

11 Appendix

Table 2: Description of songs' attributes

Variable Name	Type	Description
Name	Qualitative	Name of Track
Album	Qualitative	Name of Album
Artist	Qualitative	Name of Artist
Length	Qualitative	Duration of Track
Danceability	Quantitative	Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability and beat strength. A value of 0.0 is the least danceable and 1.0 is the most danceable.
Acousticness	Quantitative	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
Energy	Quantitative	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy.
Instrumentalness	Quantitative	Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal".
Liveness	Quantitative	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live.
Loudness	Quantitative	The overall loudness of a track in decibels (dB). Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typically range between -60 and 0 db.
Speechiness	Quantitative	Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show), the closer to 1.0 the attribute value.
Tempo	Quantitative	The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a piece and derives from the average beat duration.
Time signature	Qualitative	An estimated time signature. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).

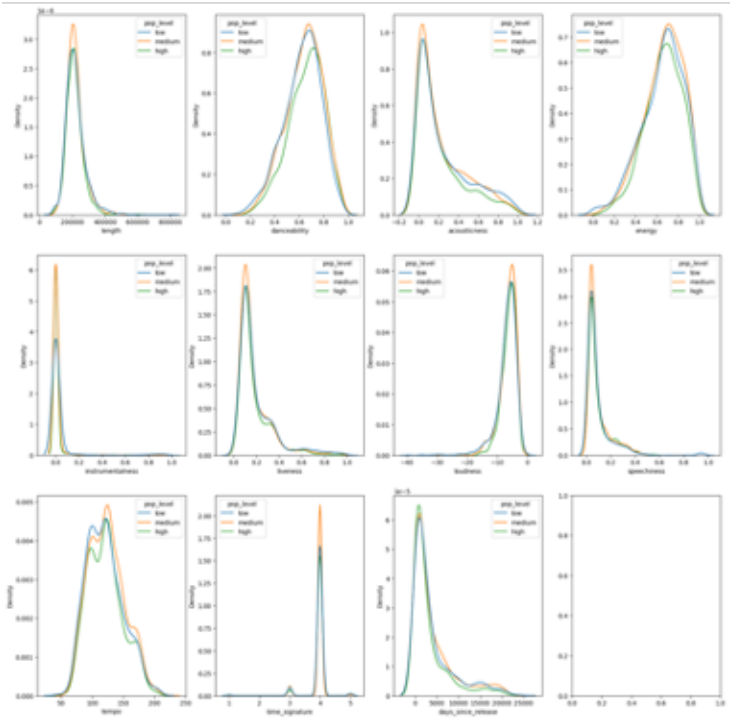


Figure 17: Kernel Density plot of target vs predictors

```
lasso_params=pd.Series(results_lasso.params)
columns_lasso=list(lasso_params[lasso_params!=0].index)
columns_lasso
```

```
['const',
 'loudness',
 'tempo',
 'popularity_meta',
 'album_avg_pop_other',
 'imputed_album_avg_pop_other',
 'danceability_speechiness_feat_3.0',
 'length_speechiness_feat_999.0',
 'instrumentalness_time_signature_feat_2.0',
 'instrumentalness_time_signature_feat_999.0',
 'length_tempo_feat_2.0',
 'length_tempo_feat_999.0',
 'days_since_log']
```

Figure 18: Variables selected after fitting LASSO regression.

29	3.002400e+15	instrumentalness_time_signature_feat_3.0
18	2.929742e+01	time_signature_4
17	2.511567e+01	time_signature_3
38	1.378926e+01	followers_root
14	1.282924e+01	artist_avg_pop_other
11	1.042413e+01	followers
15	9.574925e+00	album_avg_pop_other
6	7.371623e+00	loudness
7	6.443019e+00	speechiness
23	6.151415e+00	danceability_speechiness_feat_3.0
13	5.522221e+00	popularity_count_album
19	5.393267e+00	time_signature_5
16	4.582140e+00	imputed_album_avg_pop_other
2	4.349558e+00	acousticness
10	3.706973e+00	popularity_meta
20	3.042931e+00	liveness_speechiness_feat_2.0
3	3.027165e+00	energy
4	2.927711e+00	instrumentalness
37	2.873046e+00	instrumentalness_acousticness
35	2.671437e+00	acousticness_loudness
0	2.209617e+00	length
34	2.056073e+00	days_since_log
9	2.053527e+00	days_since
5	1.706838e+00	liveness

Figure 19: VIF before fitting LASSO regression.

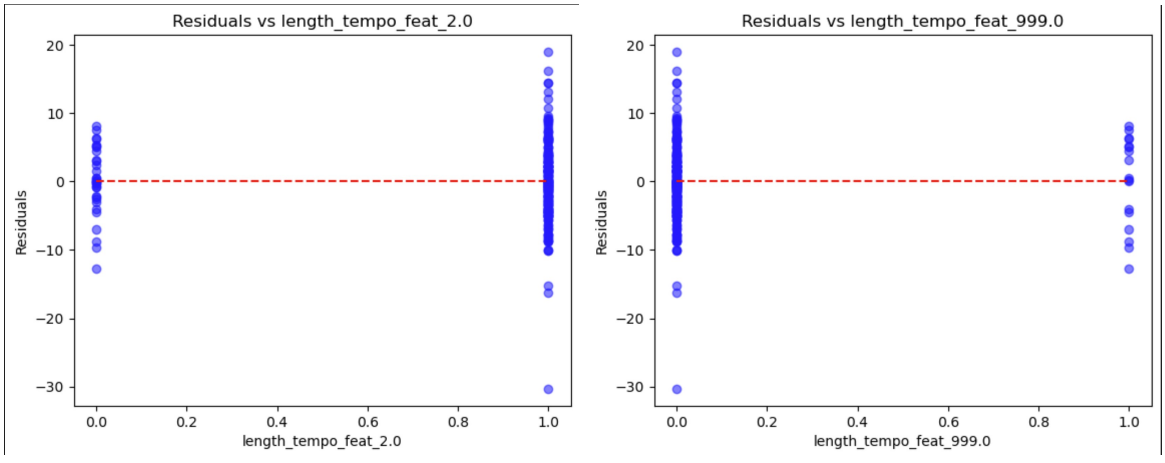


Figure 20: Residuals vs imputed_album_avg_pop_other and instrumentalness_time_signature_feat_2.0

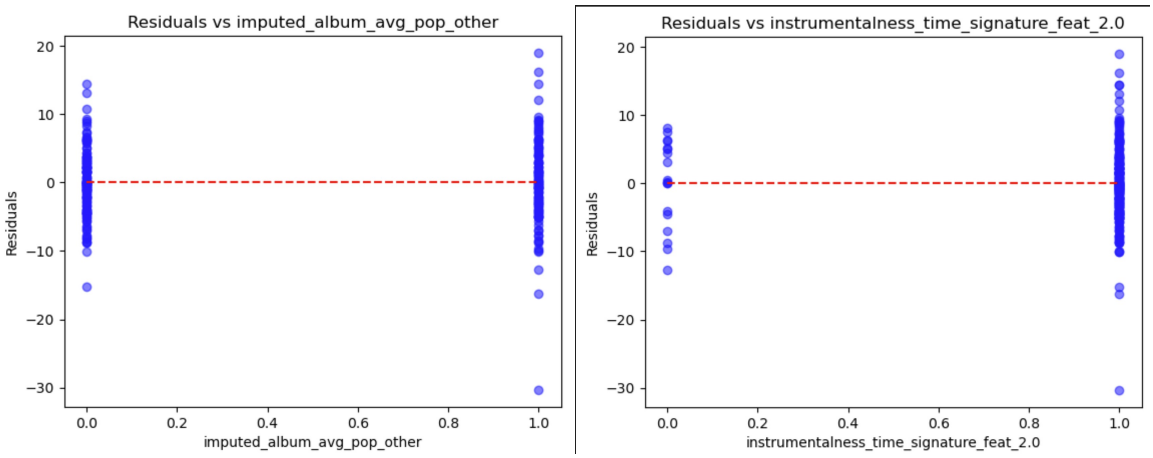


Figure 21: Residuals vs imputed_album_avg_pop_other and instrumentalness_time_signature_feat_2.0

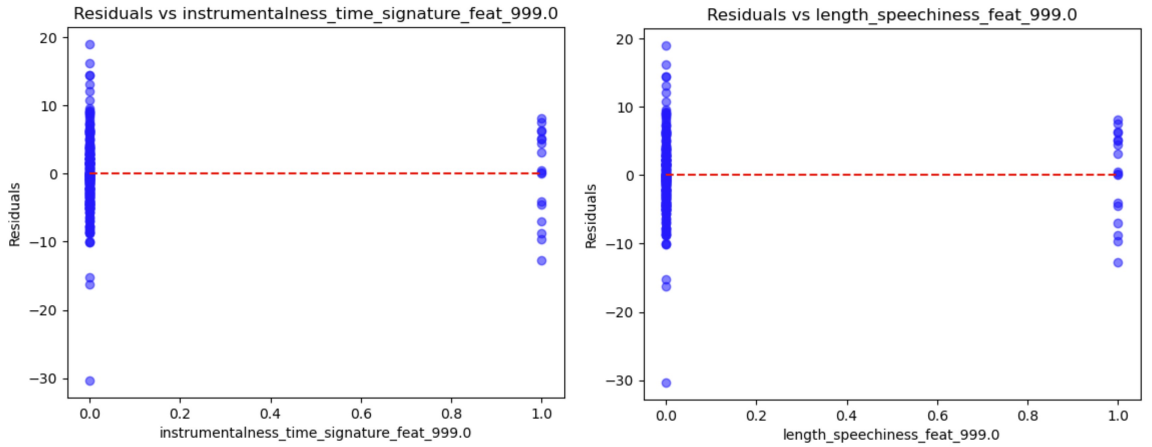


Figure 22: Residuals vs imputed_album_avg_pop_other and instrumentalness_time_signature_feat_2.0

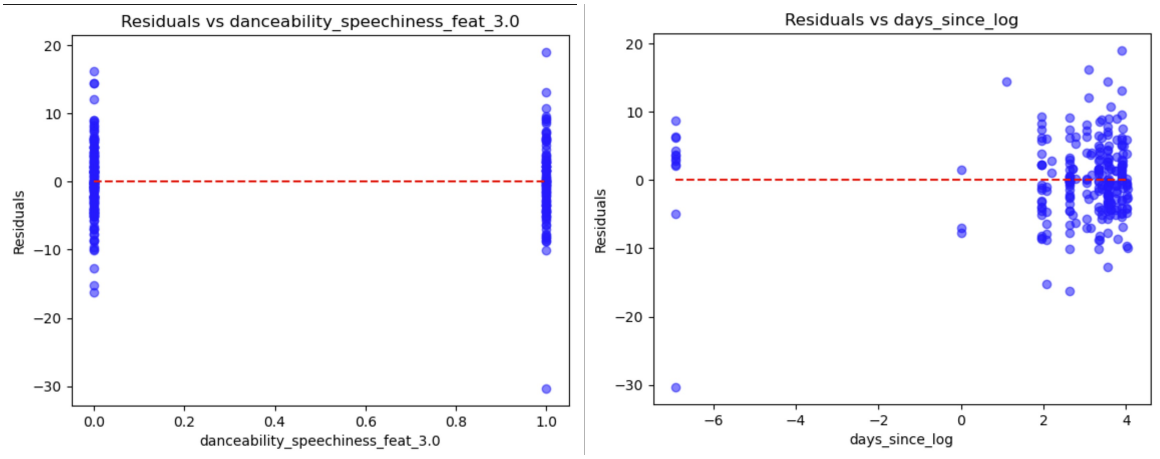


Figure 23: Residuals vs danceability_speechiness_feat_3.0 and days_since_log