Popularify - A ML prediction tool for song's popularity

Andrea D'Arpa - 0000983830
andrea.darpa@studio.unibo.it

1 Abstract

In today's world, music is mostly consumed through streaming services such as Spotify, Deezer, Apple Music, etc. Furthermore, with the advancement of technology, anyone can set up their own home studio within the walls of their bedroom. In this scenario, we find a multitude of new artists who want to approach the release of their songs on the aforementioned platforms, but they find themselves having to navigate the process without being able to rely on a record label that would oversee the process and help them evaluate their work. In this study, we aim to provide a predictive model to calculate a binary value that indicates the likelihood of the song itself becoming popular, by analyzing its characteristics.

2 Introduction

The idea behind this study is to leverage the Spotify API to analyze the characteristics of a song and relate them to the number of streams, in order to algorithmically find a pattern that determines, as closely as possible to reality, whether a track has the potential to become a hit (using industry terminology).

Of course, there are numerous relevant factors to consider in relation to popularity in order to better understand what the average listener looks for in a song.

The characteristics we will analyze are solely related to music theory and not tied to a specific artist, aiming to find a common ground that correlates the attributes of a song with its popularity.

These characteristics (that we will explore in the next section) will be correlated with the popularity value of the analyzed song in order to derive their determinancy.

For example, given that danceability is the feature that explains how much a song is suitable for dancing to it, what is the relationship between a song's danceability and his popularity? Can we extract a pattern indicating that more danceable tracks tend to rank higher on the charts of most listened-to songs?

In this study, I plan to utilize two machine learning algorithms trained on a rich and well structured dataset finded on Kaggle[1]. I will test different two models that were not considered in the study that inspired this research [2].

The referenced study analyzed the career of an Italian artist, Luciano Ligabue, using Generalized Linear Mixed Models (GLMM). However, I believe that a linear model is not the most suitable for this type of analysis, considering the subject matter. After analyzing the dataset and identifying its characteristics, I will attempt to demonstrate how a non-linear model (in this case, I have chosen decision trees) outperforms various linear models.

For a more comprehensive data collection, I have explored models other than GLMM. GLMM performances can be found in the study [2].

Analyzing the results, I expect that, after examining the model's characteristics, I will be able to derive a predictive model related to the genre of a song.

3 Data Overview and Analysis

3.1 Overview

The previously introduced dataset was used only for training, testing and so for a first evaluation. Then, the produced application can be used with every song present on spotify database simply copying and pasting the song's uri to the application.

First, we want to analyze and discover some interesting facts about our dataset.
First of all, let's give some useful information about this dataset.

- This dataset has 130663
- We have 17 attributes for each song
- Our dependent variable is the **popularity** that ranges between 0 and 100

Of this 17 values, we will consider 13 numerical values that the spotify api give us about a song:

- acousticness: A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
- danceability: Describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
- duration_ms: The duration of the track in milliseconds.
- energy: 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. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
- instrumentalness: 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". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.
- Key: The key the track is in. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C sharp / D flat, 2 = D, and so on. If no key was detected, the value is -1.
- liveness: Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.
- loudness: The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. 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.
- mode: Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
- speechiness: Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.
- tempo The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.
- time_signature: An estimated time signature. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure). The time signature ranges from 3 to 7 indicating time signatures of "3/4", to "7/4".
- valence: A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

3.2 Exploratory Data Analysis

Firstly, i tried to understand how the popularity values were distributed inside the dataset.

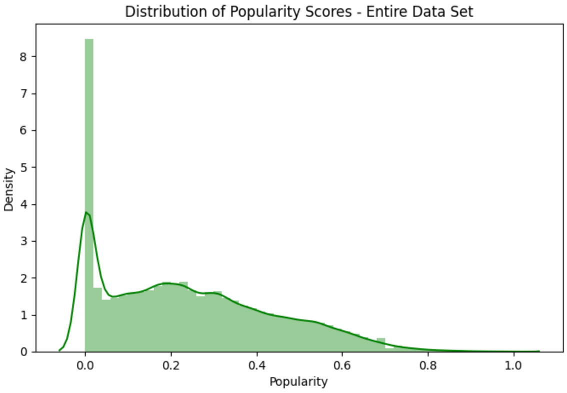

Figure 1: Popularity distribution on the dataset. Generated with python library plotlib

We can notice how less than 10% of our songs have popularity score grater than 55.
Going further i further realized that (as stated in the introduction) linear regression could not be able to solve the problem I am facing. This is due to the fact that many of our features seems not having correlation with our popularity score.
Ultimately, i wanted to understand if we could have some multicollinearity, and so, i tried to put in a plotlib heatmap variables to analyze their dependancy.
I could notice that some pairs like energy and loudness were fairly correlated, but for the sake of our analysis, we have to analyze the column correlating with popularity, and in that case, our results are far from promising for our linear regression model.

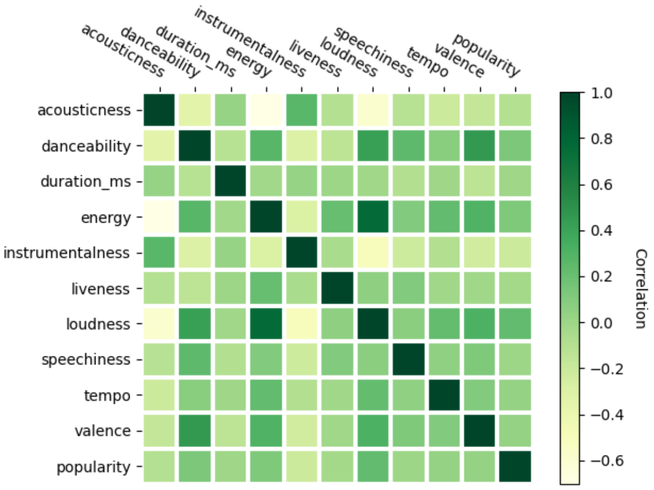

Figure 2: Heatmap of our numerical values in the dataset.

To better see this i also printed some scatterplots of our features in correlation with popularity, to better visualize how low this correlations are. You can find the plots at the appendix of this paper.

3.3 Exploratory Data Analysis Summary

To summarize this data exploration, I would like to highlight some of the most relevant statistics:

- Distribution of dependent variable is unbalanced, and this makes harder for a model to predict songs with a very high popularity value because the dataset offers less popular songs to train on

- We don't have Null values, but we have many 0's that can cause some problem while training
- We found some potential multicollinearity between valence and energy as the heatmap shows

4 Results

In this section we are going to illustrate what we have found applyng linear and then non linear probability models [3] to our dataset in order to extract a prediction model.

4.1 Linear Model

I firstly analized *Linear Regression*. This model was chosen to try to predict the actual popularity value.
To do so, the dataset was split in two parts: 80% for training and 20% for testing.
By adding and removing features, trasforming them and filtering the data, I tried to achieve a good R^2 value.
After trying to deleting songs with popularity equal to 0, i could not increase by much my R^2 , that was equal to 0.18 in the first version of the Linear Regression approach (as shown in the summary table following).
I then managed to obtain my model, summarized as follows:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             popularity   R-squared:                       0.191
Model:                            OLS   Adj. R-squared:                  0.189
Method:                 Least Squares   F-statistic:                     126.2
Date:                Mon, 10 Jul 2023   Prob (F-statistic):          1.57e-307
Time:                        18:20:09   Log-Likelihood:                -32498.
No. Observations:                6976   AIC:                         6.502e+04
Df Residuals:                    6962   BIC:                         6.512e+04
Df Model:                          13
Covariance Type:            nonrobust
====================================================================================
                       coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
const               55.9249      3.494     16.004      0.000      49.075      62.775
acousticness         0.6065      1.419      0.427      0.669      -2.175       3.388
danceability        19.4563      2.258      8.617      0.000      15.030      23.882
duration_ms_std     -0.9409      0.313     -3.004      0.003      -1.555      -0.327
energy             -20.4085      2.492     -8.191      0.000     -25.293     -15.524
instrumentalness   -17.4741      1.325    -13.188      0.000     -20.071     -14.877
key                 -0.1242      0.086     -1.449      0.148      -0.292       0.044
liveness            -9.7545      2.080     -4.690      0.000     -13.832      -5.677
loudness_std         9.7507      0.594     16.418      0.000       8.586      10.915
mode                -1.7157      0.639     -2.687      0.007      -2.967      -0.464
speechiness         -3.7270      2.799     -1.332      0.183      -9.213       1.759
tempo_std            0.2213      0.314      0.705      0.481      -0.394       0.836
time_signature       0.8999      0.699      1.287      0.198      -0.471       2.271
valence            -12.8419      1.493     -8.603      0.000     -15.768      -9.916
==============================================================================
Omnibus:                      505.311   Durbin-Watson:                   2.013
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              344.724
Skew:                          -0.432   Prob(JB):                     1.39e-75
Kurtosis:                       2.338   Cond. No.                         97.0
==============================================================================
```

This results were far from being usable, as we can see in this graph showing the correlation between True and Predicted Popularity values.

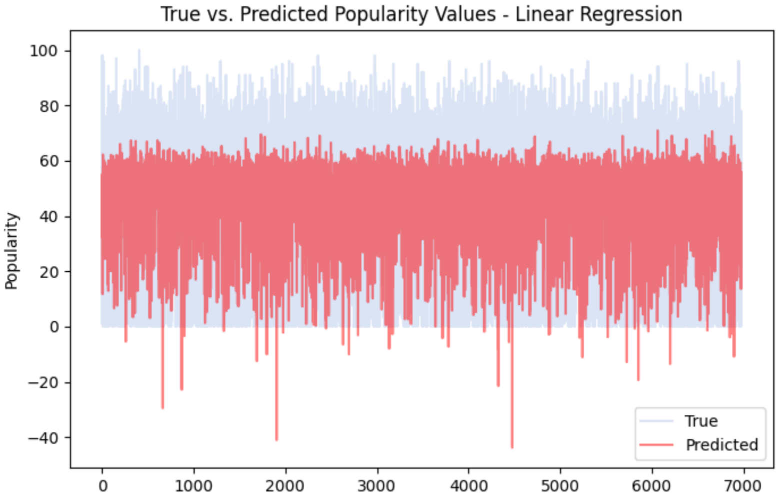

Figure 3: Model predictions results

Even though we have a not so usable model, with a really low R^2, I obtained some helpfull information about our song's features.
The most impactful features on our results appears to be **danceability**, **energy**, **instrumentalness**, **speechiness** and **valence**.

4.1.1 Undersampling

The primary challenge encountered in the initial approach stems from the highly imbalanced nature of the data. Predicting the highest and lowest popularity values becomes exceptionally challenging under these circumstances. However, upon discussing this issue with colleagues, it was collectively determined that undersampling could offer an effective solution for addressing such an unbalanced dataset.

Undersampling involves balancing the ratio between important and unimportant values of the dependent variable, thereby enabling the model to encounter a more substantial number of values that are of interest. This is achieved by initially selecting a subset of the data that comprises all records with a high popularity score. This subset is subsequently defined as the cutoff point.

Consequently, all values with a popularity score greater than or equal to the cutoff are included in the model. The data with popularity scores below the cutoff point is then randomly sampled, ensuring an equal distribution of popular and unpopular songs in the final dataset, amounting to a 50/50 split. The schematic representation of this process is depicted in the accompanying diagram.

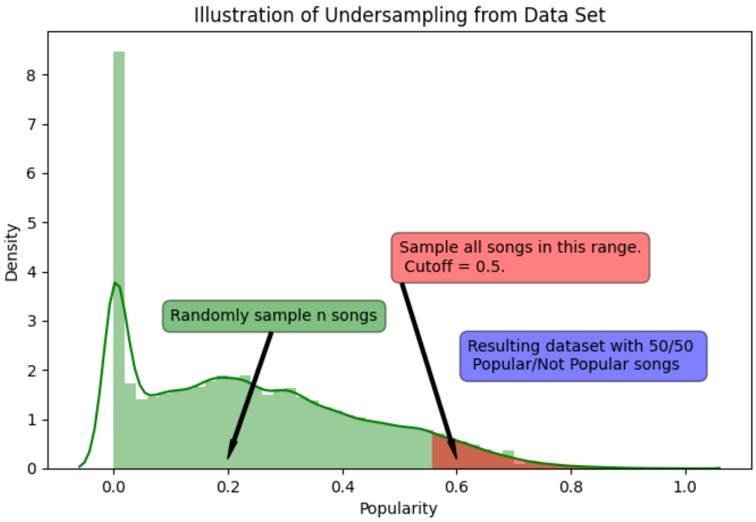

Figure 4: Model predictions results

Then I focused on maximizing my R^2 values, by experimenting with different cutoff values. To better show how the model's prediction reacted to this changes, observe this plots:

Model	R$\hat{2}$	Danceability Coeff	Danceability p-value	Danceability Coeff Range
First Linear Model	0.08	5.26	0.00	4.441-6.079
Cutoff = 55	0.14	16.07	0.00	13.833-18.318
Cutoff = 65	0.19	19.45	0.00	15.030-23.882
Cutoff = 75	0.3	46.00	0.00	36.415-55.594
Cutoff = 85	0.38	75.69	0.00	49.217-102.182
Cutoff = 90	0.35	70.56	0.002	26.471-114.658

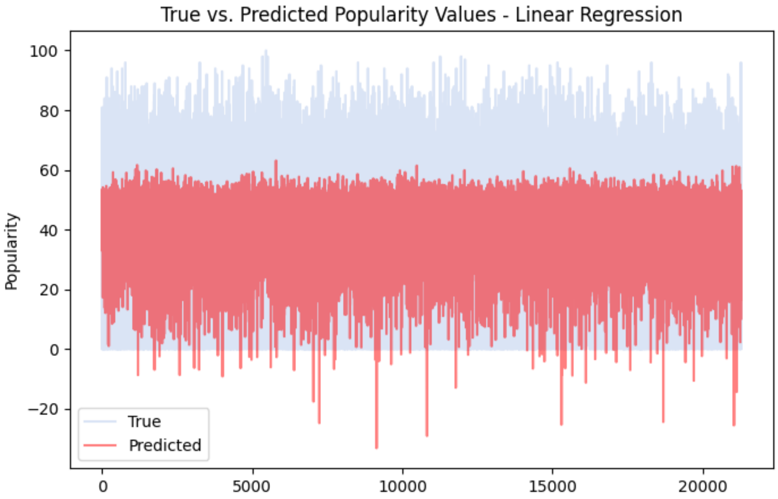

Figure 5: Model predictions results with cutoff 55

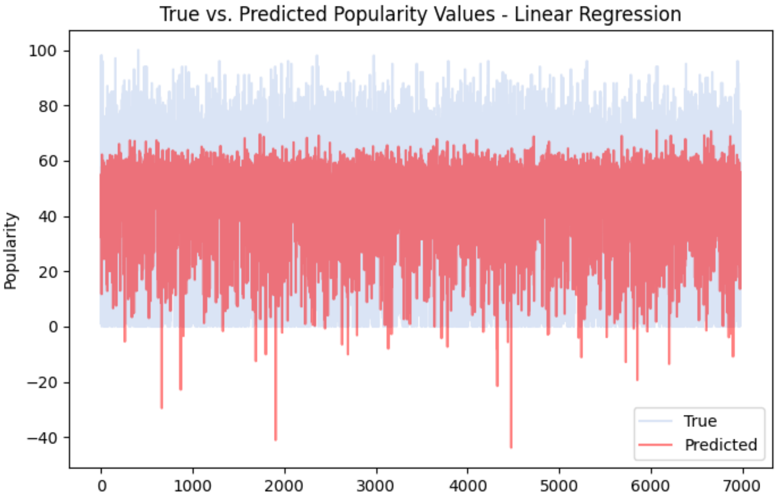

Figure 6: Model predictions results with cutoff 65

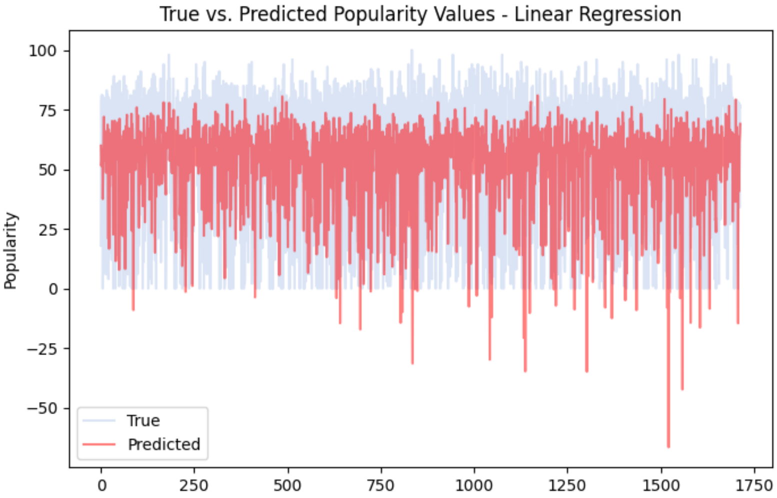

Figure 7: Model predictions results with cutoff 75

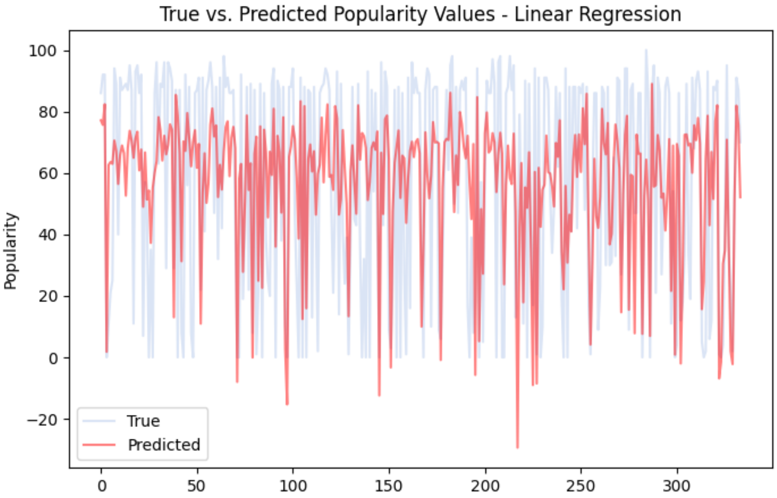

Figure 8: Model predictions results with cutoff 85

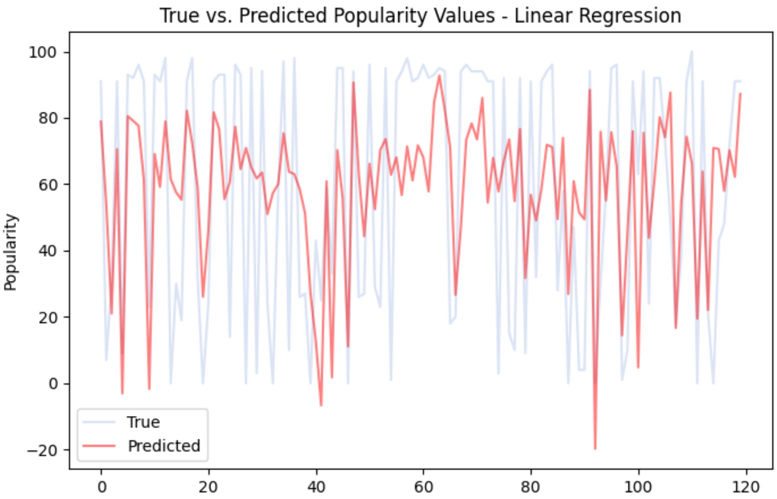

Figure 9: Model predictions results with cutoff 90

So, the linear regression model I think is the best one, is the one that has cutoff set to 85.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             popularity   R-squared:                       0.381
Model:                            OLS   Adj. R-squared:                  0.356
Method:                 Least Squares   F-statistic:                     15.15
Date:                Mon, 10 Jul 2023   Prob (F-statistic):           1.26e-26
Time:                        19:06:28   Log-Likelihood:                -1592.7
No. Observations:                 334   AIC:                             3213.
Df Residuals:                     320   BIC:                             3267.
Df Model:                          13
Covariance Type:            nonrobust
====================================================================================
                       coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
const               -4.9457     26.024     -0.190      0.849     -56.146      46.255
acousticness        11.2034      7.670      1.461      0.145      -3.886      26.293
danceability        75.6997     13.461      5.624      0.000      49.217     102.182
```

```
duration_ms_std     -4.2372      1.638     -2.587      0.010      -7.460      -1.015
energy              -3.7938     13.688     -0.277      0.782     -30.723      23.136
instrumentalness   -36.6052      7.122     -5.140      0.000     -50.617     -22.593
key                 -0.2633      0.455     -0.578      0.563      -1.159       0.632
liveness             6.8517     11.571      0.592      0.554     -15.914      29.617
loudness_std         7.3425      3.242      2.265      0.024       0.964      13.721
mode                 3.1196      3.357      0.929      0.353      -3.486       9.725
speechiness         -9.3169     17.574     -0.530      0.596     -43.891      25.257
tempo_std            1.5926      1.698      0.938      0.349      -1.748       4.934
time_signature       6.4881      5.485      1.183      0.238      -4.302      17.279
valence            -22.8526      8.709     -2.624      0.009     -39.988      -5.718
==============================================================================
Omnibus:                       28.551   Durbin-Watson:                   2.099
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               33.428
Skew:                          -0.758   Prob(JB):                     5.51e-08
Kurtosis:                       2.682   Cond. No.                         132.
==============================================================================
```

To end the Linear Model Evaluation, i computed the Root Mean Squared Error, that was in the test phase smaller than i expected, but still not auspicable:
RMSE for training set 29.513584123038008
RMSE for test set 22.84432001521449
Non the less, we had our confirmation on our most important features, that still appears to be the same of our first model.

4.2 Non Linear Model: Logistic Regression

For this second part, i used a slightly different approach, trying to give a binary answer to the question: is this song popular or not?. So, i added some binary popularity values using our cutoff, giving us a value of 0 if the song is unpopular (popularity score minor than cutoff) and 1 if the song is popular (popularity score grater equal than 0).
Considering the importance of AUC, Accuracy, Precision and Recall [4] in the evaluation of a Logit model, i decided to see if also in this case, our cutoff would impact those values in any ways.
The above figure illustrates that this is indeed the case.

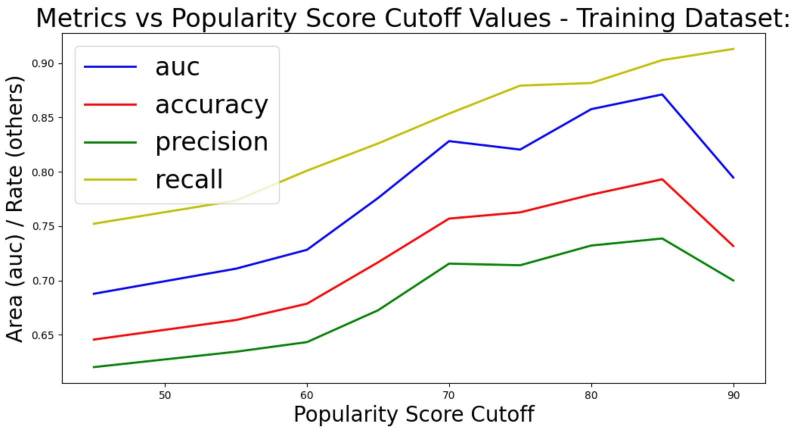

Figure 10: Plot showing how our cutoff changes metric values

After careful consideration, I determined that a cutoff of 85 would be appropriate. This decision was based on observing a decline in most metrics beyond this threshold. Additionally, the linear regression analysis revealed a diminishing significance of the features as the cutoff value increased. There i show the results confusion matrix for both Training and Test datasets.

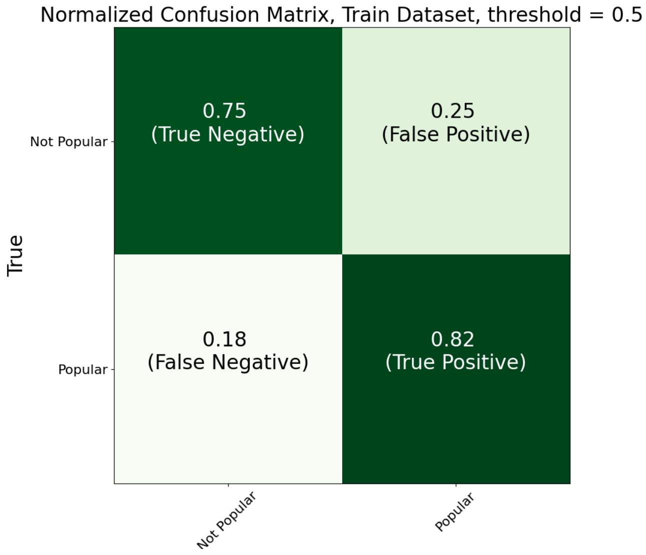

Figure 11: Plot showing how our cutoff changes metric values

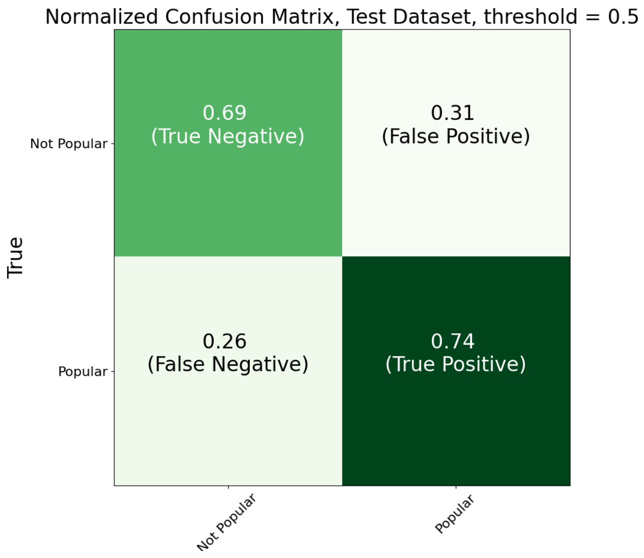

Figure 12: Plot showing how our cutoff changes metric values

I then applied standard equations to evaluate a Logit model: AUC (Area Under the ROC Curve):

$$AUC = \text{Total area under the ROC curve}$$

Accuracy:

$$\text{Accuracy} = \frac{\sum \text{True Positive} + \sum \text{True Negative}}{\sum \text{Total Population}}$$

Precision:

$$\text{Precision} = \frac{\sum \text{True Positive}}{\sum \text{Predicted Condition Positive}}$$

Recall / TPR / Sensitivity:

$$\text{Recall / TPR / Sensitivity} = \frac{\sum \text{True Positive}}{\sum \text{Condition Positive}}$$

Using a threshold value of 0.5 to divide popular from unpopular songs i obtained my model performance table.

Table 1: Model Performance

	Train	Test heightAccuracy
0.78	0.72 heightRecall	0.82
0.74 heightPrecision	0.75	0.76 heightAUC
0.84	0.82 height	

In the broader context, danceability continues to exhibit significant importance as a feature in predicting the level of popularity. Additionally, energy and instrumentalness have emerged as notable predictors, albeit with negative coefficients.
This suggests that, holding all other factors constant, an increment of one unit in either energy or instrumentalness corresponds to a decrease in the likelihood of a song achieving popularity.
Also, loudness also appears as a prominent indicator, reflecting the true dB maximum output at which most ”popular” songs are typically mastered and released.

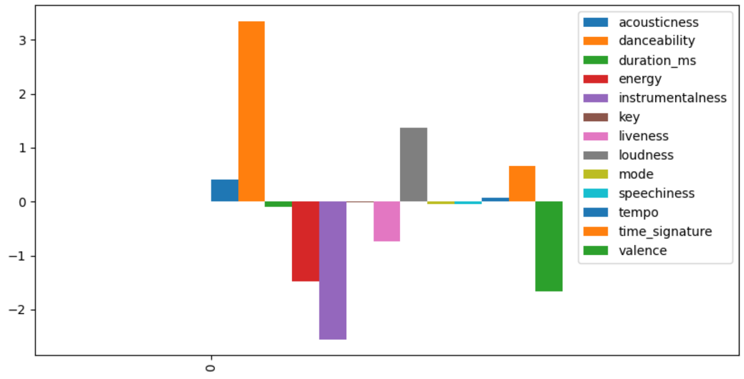

Figure 13: Plot showing final coefficient values.

5 Conclusions

5.1 Discussions

To summarize, this paper aimed to develop a predictive model for determining the likelihood of a song becoming popular based on its characteristics. The study utilized the Spotify API and a dataset containing various musical features to train and evaluate different models.

Initially, a linear regression model was used to predict the actual popularity value of songs. However, due to imbalanced data and weak correlations between the features and popularity, the results were unsatisfactory. To address this, different approaches were explored, including undersampling and experimenting with different cutoff values. Ultimately, a linear regression model with a cutoff of 85 provided the best results, achieving an R-squared value of 0.381 for the training set. However, the model still had limitations, as indicated by relatively high RMSE values.

To overcome these limitations, a non-linear approach using logistic regression was adopted to predict whether a song would be popular or not, based on binary popularity values derived from the cutoff. The logistic regression model performed better, with an accuracy of 0.72 for the test set. Noteworthy predictors of popularity included danceability, energy, instrumentalness, and loudness.

5.2 Future Works

The obtained results are decent, and there is potential for practical applications from the perspective of the music industry for such a predictive algorithm. However, to make this work truly usable, it can be asserted that there is a need for testing different models to arrive at more robust evaluations of the predictive model. Also, finding a way to obtain the values of song features before uploading them to the Spotify platform is crucial to find a real application inside of music production studios.

6 Appendix

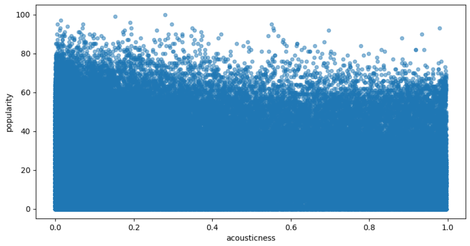

Figure 14: Scatterplot of popularity vs. acousticness

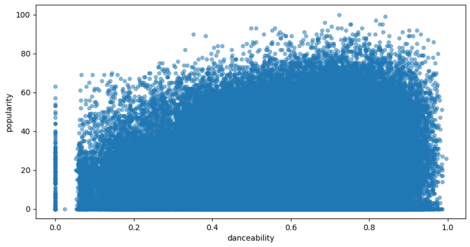

Figure 15: Scatterplot of popularity vs. danceability

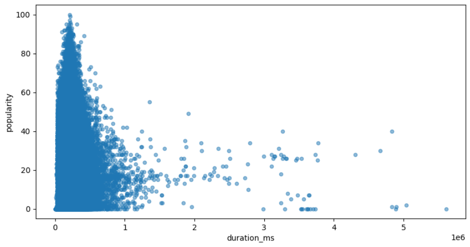

Figure 16: Scatterplot of popularity vs. duration

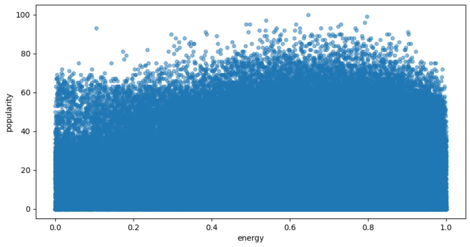

Figure 17: Scatterplot of popularity vs. energy

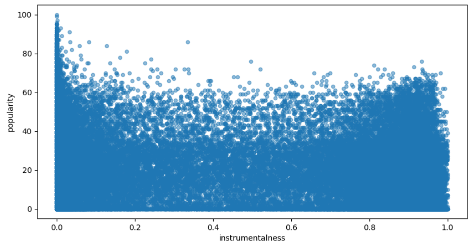

Figure 18: Scatterplot of popularity vs. instrumentalness

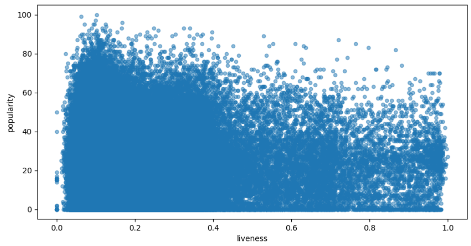

Figure 19: Scatterplot of popularity vs. liveness

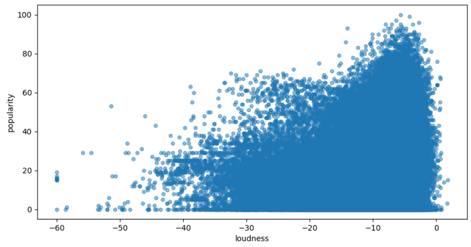

Figure 20: Scatterplot of popularity vs. loudness

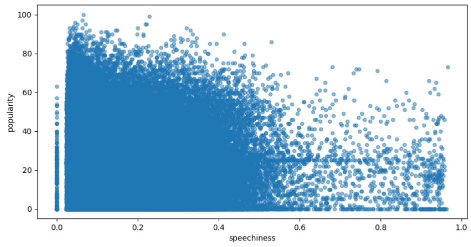

Figure 21: Scatterplot of popularity vs. speechiness

References

[1] Spotify api - april 2019. /https://www.kaggle.com/code/kerneler/starter-spotify-audio-features-ba0befd6-6/input.

[2] Mariangela Sciandra and Irene Carola Spera. A model-based approach to spotify data analysis: a beta GLMM. *Journal of Applied Statistics*, 49(1):214–229, August 2020.

[3] Richard Breen, Kristian Karlson, and Anders Holm. Interpreting and understanding logits, probits, and other nonlinear probability models. *Annual Review of Sociology*, 44:1–16, 05 2018.

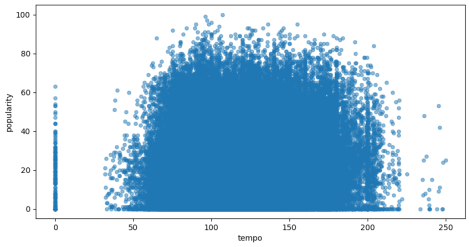

Figure 22: Scatterplot of popularity vs. tempo

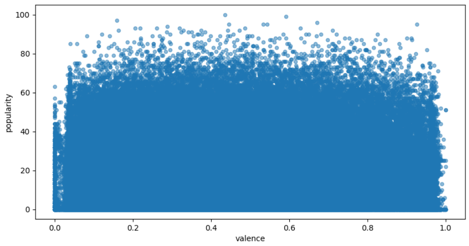

Figure 23: Scatterplot of popularity vs. valence

[4] Andre M. Carrington, Douglas G. Manuel, Paul W. Fieguth, Tim Ramsay, Venet Osmani, Bernhard Wernly, Carol Bennett, Steven Hawken, Olivia Magwood, Yusuf Sheikh, Matthew McInnes, and Andreas Holzinger. Deep ROC analysis and AUC as balanced average accuracy, for improved classifier selection, audit and explanation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):329–341, January 2023.