CSCI 183 Final Project Proposal

Group Members:
Sayan Jain
Sahana Kumar
Shivani Dharanipragada
Emma Wegner

Can We Use Data on a Song's Metrics to Predict a Song's Popularity?

I. Abstract

Using a set of 114,000 songs and their key features, after data cleaning, we were able to train a supervised learning algorithm to predict popularity based on 81,000 songs with an accuracy of 80.12% and a F1 Score of 84.68%. We determined that based on the techniques available to us, the best model was a **Logistic Regression** model mapping **'danceability', 'loudness', 'speechiness', 'instrumentalness', and 'genre'** to the target of **'popularity'**.

II. Introduction & Background

In the US, the music industry generates billions of dollars yearly. In the first half of 2024 alone, it generated around 8.7 billion dollars[1]. More popular songs likely generate more revenue, making the ability to predict a song a potentially lucrative endeavor for artists, music labels, and streaming services alike. Quantifying popularity, however, like quantifying any subjective characteristic, is inherently very difficult to achieve in an objective manner. Several studies have been conducted to attempt to predict song popularity using different aspects of the song with various datasets. One study attempted to connect musical homophily, or "the tendency that people who are socially linked also share musical tastes", to a song's popularity[2]. Another used six machine learning algorithms, sentiment analysis, and metadata to derive correlations[3]. One Cornell study even posed questions about the correlations between the year a song was released, the genre of the artist, tempo, loudness, and a song's popularity[4].

Our group's primary question was if predicting a song's popularity is even possible. To tackle this question, our group made use of a Spotify dataset[5] of over 114,000 songs, alongside key identifiers of what makes a song unique, to attempt to predict the popularity of a song based on its features. Some of these features, like tempo, key, and genre, are cut-and-dry in their interpretation. Others, like danceability and energy, are much more subjective based on the listener. We planned to see if these subjective features have an actual impact on a song's popularity, our target variable.

[1] https://www.riaa.com/wp-content/uploads/2024/08/RIAA-Mid-Year-2024-Revenue-Report.pdf
[2] https://www.nature.com/articles/s41598-024-58969-w
[3] https://ieeexplore.ieee.org/abstract/document/9633884
[4] https://pages.github.coecis.cornell.edu/info2950-s23/project-fabulous-raichu/eda.html
[5] https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset/data

The Spotify dataset our group used is riddled with incomplete rows, duplicates, and random noise. This report will outline later on how we preprocessed our data in order to streamline analysis and reduce confusion. While training and testing our model, our features were: ‘track_id’ (string), ‘artists’ (string), ‘album_name’ (string), ‘track_name’ (string), ‘duration_ms’ (int), ‘explicit’ (bool), ‘danceability’ (int), ‘energy’ (int), ‘key’(int), ‘loudness’ (float), ‘mode’ (int), ‘speechiness’ (float), ‘acousticness’ (float), ‘instrumentalness’ (float), ‘liveness’ (float), ‘valence’ (float), ‘tempo’ (float), ‘time_signature’ (int), ‘track_genre’ (string). Our **target** was ‘popularity’.

III. Design

To answer our question, we first decided to try out both classification (Logistic Regression and SVC) and regression (Linear Regression) models to see which method was more accurate. We further decided to follow three main steps: data preprocessing, data visualization, and model construction.

A. Data Preprocessing

The first step we decided to take was to clean our data. We decided to remove incomplete rows and then duplicate songs. We defined duplicate songs as songs that had the same song title and artist. To our dataset, these songs may appear different because they appear on different albums or for other reasons. However, many artists release songs twice—once as a single, and again as a part of an album—and thus, we determined it would be appropriate to remove those songs as duplicates.

We then decided that after preprocessing our data, we would then separate our target feature, ‘popularity’, into a binary and categorical column. We made this decision after looking closely at the ‘popularity’ column. ‘Popularity’ rated songs on a scale from 1 to 100, with ‘1’ indicating ‘not popular’ and ‘100’ indicating ‘popular’. In class, we experimented with two kinds of models: those that yield a binary answer of true (1) or false (0) and models that choose between several categories. We did not want to make our model predict a song’s popularity on a scale of 1 to 100, as that would yield a confusion matrix of size 100 x 100. However, our target being categorical seemed useful to experiment with as well. Thus, to challenge ourselves, we decided to run each model twice:

First, we decide to make a column with the target being a classic binary of ‘popular’ or ‘not popular’:

‘Not popular’/0: ‘Popularity’ < 25
‘Popular’/1: ‘Popularity’ >= 25

Second, we decided to make a column with the target being categorized on popularity 0, 1, 2, 3:

0: 0 <= popularity < 25
1: 25 <= popularity < 50
2: 50 <= popularity < 75
3: 75 <= popularity <=100

B. Data Visualizations

After deciding to run models for both a binary and categorical target, we decided that we should visualize different features to see how they would affect popularity. We kept in mind that even with the sheer size of our dataset, having a very small percentage of outliers could still have an impact on the graph itself.

We decided to first visualize songs based on features that we inherently thought would truly affect a song's popularity while ignoring features that we thought would have no impact. We also decided to make a correlation matrix for more clarification on how important certain features affect a song's popularity.

C. Model Construction

We decided that after visualizing our data, we'd use that data to either decide on which features to construct our model off of, or create sets of features to try constructing models with. We also further decided to run Linear Regression, Logistic Regression, and SVC models. From those models, we knew we would meet our **end goal**, which was *to find the best model that yields the highest accuracy and F1 Score and works best to predict song popularity* (see 'Implementation' and 'Results' to learn more about the exact implementation of the models, the features used, and the results).

IV. Implementation

A. Data Preprocessing

First, we preprocessed our data. We dropped all of the rows with missing data and removed all duplicates, leaving only one of each song. When we removed all songs with the same artist and song title, we reduced our dataset size from 114,000 to around 81,000 songs—a clear sign of how noisy our dataset was to begin with. Luckily, trimming down our dataset did not make it too small to work with.

After trimming down our dataset, we split our target variable, ‘popularity’, from a numerical feature into two new columns: one binary and one categorical. The details of how we split our target variable and why are covered above, in the ‘Implementation’ section.

B. Data Visualization

Next, we visualized song popularity based on features we thought, on their face, would affect a song’s popularity, including genre, danceability, and explicitness. We ignored features that, on their face, appeared unlikely to have an impact, such as key or valence. Here are some examples of the data visualizations we generated based on superficial judgments of feature importance, a few of which seemed to correlate somewhat with average popularity:

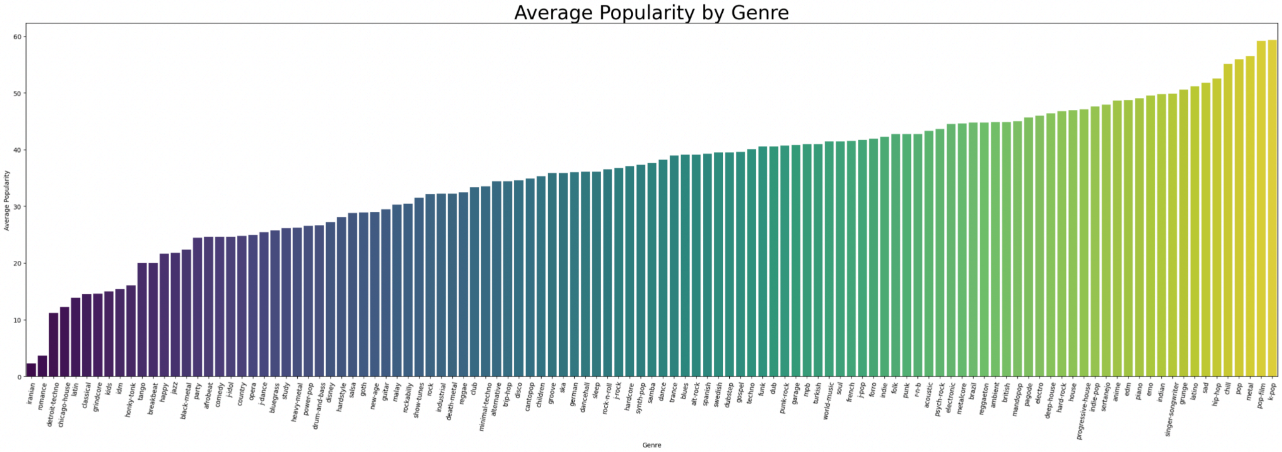


Fig 1 (Average Popularity Based on Genre)

As can be seen, the genre of a song seems to have a large impact on popularity, as some genre's average popularity was a lot higher than other data. For example, pop-films had a popularity average of around 60, while Iranian had a popularity average of less than 5.

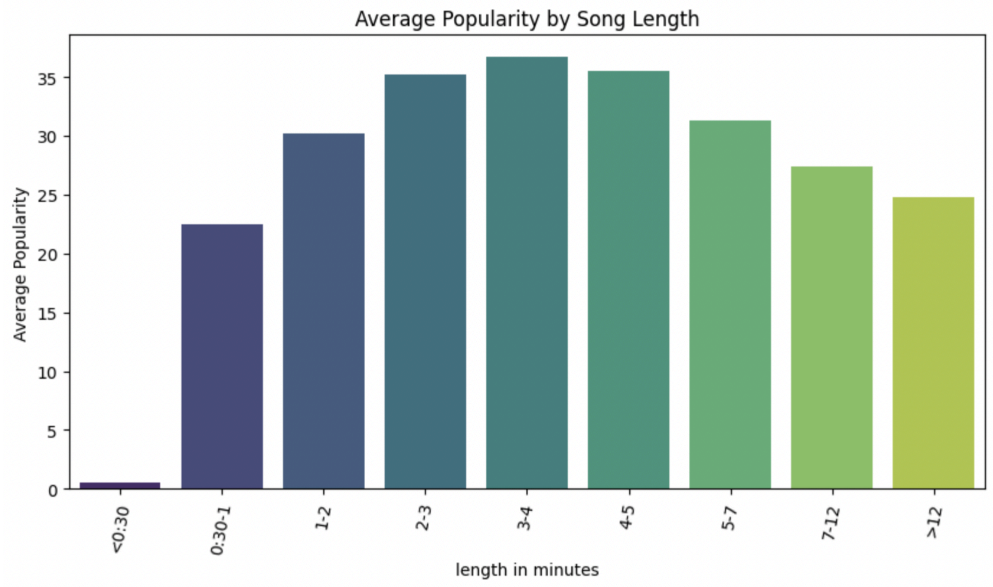


Fig 2 (Average Popularity Based on Song Length)

Song Length seemed to have a non-linear impact on song popularity, with average popularity peaking at a mid-length, or 3-4 minutes.

While the two features outlined above did indeed seem to have some level of correlation with average popularity, there were other features that had no visible correlation. Consider the following visualization of danceability and explicitness' correlation with song popularity:

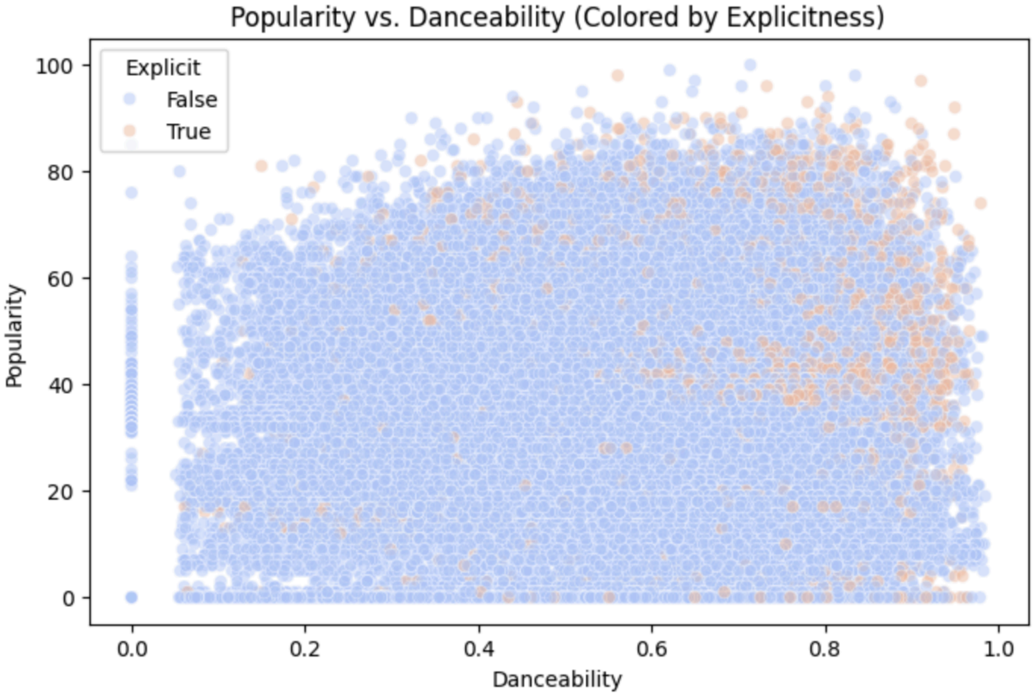
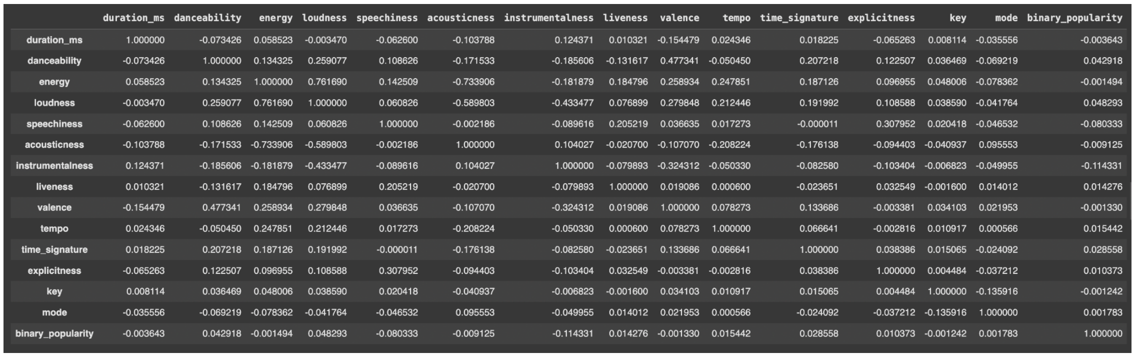


Fig 3 (Average Popularity Based on Danceability and Explicitness)

There seems to be no visible and significant trend between popularity, explicitness, and danceability, at least on the data's face.

After visualizing these features and their correlation with song popularity, we then created a correlation matrix for our numerical features.



	duration_ms	danceability	energy	loudness	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature	explicitness	key	mode	binary_popularity
duration_ms	1.000000	-0.073426	0.058523	-0.003470	-0.062600	-0.103788	0.124371	0.010321	-0.154479	0.024346	0.018225	-0.065263	0.008114	-0.035556	-0.003643
danceability	-0.073426	1.000000	0.134325	0.259077	0.108626	-0.171533	-0.185606	-0.131617	0.477341	-0.050450	0.207218	0.122507	0.036469	-0.069219	0.042918
energy	0.058523	0.134325	1.000000	0.761690	0.142509	-0.733906	-0.181879	0.184796	0.258934	0.247851	0.187126	0.096955	0.048006	-0.078362	-0.001494
loudness	-0.003470	0.259077	0.761690	1.000000	0.060826	-0.589803	-0.433477	0.076899	0.279848	0.212446	0.191992	0.108588	0.038590	-0.041764	0.048293
speechiness	-0.062600	0.108626	0.142509	0.060826	1.000000	-0.002186	-0.089616	0.205219	0.036635	0.017273	-0.000011	0.307952	0.020418	-0.046532	-0.080333
acousticness	-0.103788	-0.171533	-0.733906	-0.589803	-0.002186	1.000000	0.104027	-0.020700	-0.107070	-0.208224	-0.176138	-0.094403	-0.040937	0.095553	-0.009125
instrumentalness	0.124371	-0.185606	-0.181879	-0.433477	-0.089616	0.104027	1.000000	-0.079893	-0.324312	-0.050330	-0.082580	-0.103404	-0.006823	-0.049955	-0.114331
liveness	0.010321	-0.131617	0.184796	0.076899	0.205219	-0.020700	-0.079893	1.000000	0.019086	0.000600	-0.023651	0.032549	-0.001600	0.014012	0.014276
valence	-0.154479	0.477341	0.258934	0.279848	0.036635	-0.107070	-0.324312	0.019086	1.000000	0.078273	0.133686	-0.003381	0.034103	0.021953	-0.001330
tempo	0.024346	-0.050450	0.247851	0.212446	0.017273	-0.208224	-0.050330	0.000600	0.078273	1.000000	0.066641	-0.002816	0.010917	0.000566	0.015442
time_signature	0.018225	0.207218	0.187126	0.191992	-0.000011	-0.176138	-0.082580	-0.023651	0.133686	0.066641	1.000000	0.038386	0.015065	-0.024092	0.028558
explicitness	-0.065263	0.122507	0.096955	0.108588	0.307952	-0.094403	-0.103404	0.032549	-0.003381	-0.002816	0.038386	1.000000	0.004484	-0.037212	0.010373
key	0.008114	0.036469	0.048006	0.038590	0.020418	-0.040937	-0.006823	-0.001600	0.034103	0.010917	0.015065	0.004484	1.000000	-0.135916	-0.001242
mode	-0.035556	-0.069219	-0.078362	-0.041764	-0.046532	0.095553	-0.049955	0.014012	0.021953	0.000566	-0.024092	-0.037212	-0.135916	1.000000	0.001783
binary_popularity	-0.003643	0.042918	-0.001494	0.048293	-0.080333	-0.009125	-0.114331	0.014276	-0.001330	0.015442	0.028558	0.010373	-0.001242	0.001783	1.000000

Fig 4 (Correlation Matrix)

C. Feature Choice

As is clearly indicated by the correlation matrix, there is not an especially strong correlation between any of the features and the target. Notably, however, after running a linear regression model on genre and popularity, a linear relationship between the two features could be seen. Thus, although we disregarded the categorical features that were strings as there was no clear way to translate a 'song title' or 'artist' numerically, we chose to include genre in our model by turning it into a numerical feature. We ordinally encoded 'genre' given its significant correlation with 'popularity'. Genres that correlated more with 'popularity' were given a higher numerical value than genres with lower correlations to 'popularity'.

D. Model Training

Our data visualization and confusion matrix showed only one feature with an absolute value correlation above 0.1 ('instrumentalness'), indicating most features had a weak correlation with 'popularity'. However, to ensure our intuitions were correct, we began training models to use features to predict 'popularity'.

We trained models using four sets of features:

1. **Numerical**: All features with a data type of 'int' or 'float'.
2. **Limited**: Features that met or went above the correlation (with 'popularity') threshold of 0.04; these features are: 'danceability', 'loudness', 'speechiness', 'instrumentalness'.
3. **Genre Included**: 'Numerical' features and 'Genre' together.

4. **Limited Features & Genre (LFG)**: ‘Limited’ features and ‘Genre’ together.

For each set, we trained three models: a traditional linear regression model, a logistic regression model, and a SVC model. The linear regression model performed poorly for the most part, while the SVC and logistic regression models performed comparatively well based on the feature sets used. Numeric metrics on all the models, including the judgment of which model performed best, can be seen in the ‘Results’ section of this report.

V. Results

Below are several tables of all the multi-class models we tried and their accuracy and F1 scores. Please note that F1 scores are not possible when our target is categorical as it results in a 4x4 matrix. Additionally, note that we did not have to make our data binary or categorical for linear regression because the model predicts the original numerical value of the target which is a number from 0-100. Finally, please note the following ‘feature sets’, defined again for convenience:

1. **Numerical**: All features with a data type of ‘int’ or ‘float’.
2. **Limited**: Features that met or went above the correlation (with ‘popularity’) threshold of 0.04.
3. **Genre Included**: ‘Numerical’ features and ‘Genre’ together.
4. **Limited Features & Genre (LFG)**: ‘Limited’ features and ‘Genre’ together.

Linear Regression:

Model	**Target Binary or Categorical**	**Accuracy Score**	**F1**
Linear Regression (Numerical)	*Numerical*	5.97%	1.02%
Linear Regression (Limited)*	*Numerical*	4.27%	0.69%
Linear Regression (Genre Included)**	*Numerical*	36.34%	3.14%
Linear Regression (LFG)**	*Numerical*	36.08%	3.21%

Linear Regression predicts a number based on the original numerical values of popularity.

Logistic Regression:

Logistic Regression (Numerical)	*Binary*	64.34%	78.30%
Logistic Regression (Numerical)	*Categorical*	41.69%	N/A
Logistic Regression (Limited)*	*Binary*	65.61%	76.94%
Logistic Regression (Limited)*	*Categorical*	45.94%	N/A
Logistic Regression (Genre Included)**	*Binary*	77.53%	83.50%
Logistic Regression w/ (Genre Included)**	*Categorical*	57.80%	N/A
Logistic Regression (LFG)***	*Binary*	80.12%	84.68%
Logistic Regression (LFG)***	*Categorical*	62.06%	N/A

SVC:

SVC (Numerical)	*Binary*	64.89%	77.72%
SVC (Numerical)	*Categorical*	23.39%	N/A
SVC (Limited)*	*Binary*	64.23%	77.36%
SVC (Limited)*	*Categorical*	30.52%	N/A
SVC (Genre Included)**	*Binary*	64.16%	76.78%
SVC (Genre Included)**	*Categorical*	22.45	N/A
SVC (LFG)***	*Binary*	76.96%	84.11%
SVC (LFG)***	*Categorical*	61.31%	N/A

* These are features that met the minimum threshold of having a positive correlation of 0.04 or higher with the target. These features are: 'danceability', 'loudness', 'speechiness', 'instrumentalness'.

**These models include the feature of genre in addition to all the other features that were originally in the dataset.

***There are features that met the minimum threshold of having a positive correlation of 0.04 or higher with the target AND 'genre', which we made into an ordinal numerical feature after discovering its high positive correlation with the genre.

To answer the questions in the guidelines:

1. *Does the final report have quantitative results from learning or experimenting with your data?*
   a. Yes, see the table above in "results"
2. *Does the final report have data analysis using visualization tools? Does the report have findings for the same?*
   a. Yes and yes, see in "B. Data Visualization in III. Implementation"
3. *Does the final report have results evaluating the learning of your model? (i.e., learning curves, precision-recall, training/testing errors, ...)*
   a. Yes, see the table above in "results"
4. *Does the final report make effective use of graphs that are appropriately labeled and properly described in the document?*
   a. Yes, see above in "B. Data Visualization in III. Implementation"

V. Interpretation

To answer our ultimate question: "Can we predict whether songs are popular or unpopular based on specific features, and to what extent?" The short answer is **yes**. The long answer? It's complicated.

We were able to create models that had more features than we had ever used before in class, with a target that was not straight-forward in it's definition nor it's interpretation. As data scientists, we had to put aside our personal beliefs about what is considered a 'popular' song and what *should* a popular song have. Additionally, we used lower-level modeling techniques to predict a target that is on a scale of 0 to 100, which entails more complicated regression techniques than what we learned. So we had to apply simpler techniques to a complicated problem. This required having restrictions on the features we were using, such as disregarding all of our string features (which very well may have an impact on a song's popularity - after all, the more popular an artist, the more popular the song, right?) One primary restriction was changing our popularity from a numerical feature on a scale of 0 to 100 to a binary feature, which limited the nuance in our predictions.

From our limited lower-level techniques, we were able to determine that the best model was a **Logistic Regression** model mapping **‘danceability', 'loudness', 'speechiness', 'instrumentalness’, and ‘genre’** to the target of ‘popularity’. From this model, we were able to get an 80.12% accuracy level and a 84.68% F1 Score.