Maximus Wu
Professor Koehler
Data Bootcamp
12 May 2025

# Data Bootcamp – Final Project Writeup

**Project Link:** https://github.com/mw5707/data-bootcamp-final-maximuswu

**Introduction and Objectives of the Project:**

This project aims to explore the relationship between various musical features and track popularity. Using a dataset containing information on tempo, energy, danceability, and other audio characteristics, I wanted to:

1. Identify what factors most to making a 'hit' song
2. Evaluate the evolution of popular music, and see what characteristics of music were changing over time
3. Conduct some statistical analysis on music streaming patterns to see if there were some traits that were necessary or at least very correlated to a music's popularity.
4. Create predictive models that assess qualitative and quantitative factors of a track to predict the popularity of the song.
5. Evaluate the various models and interpret their results. From there, conclude the project with reflections and next steps.

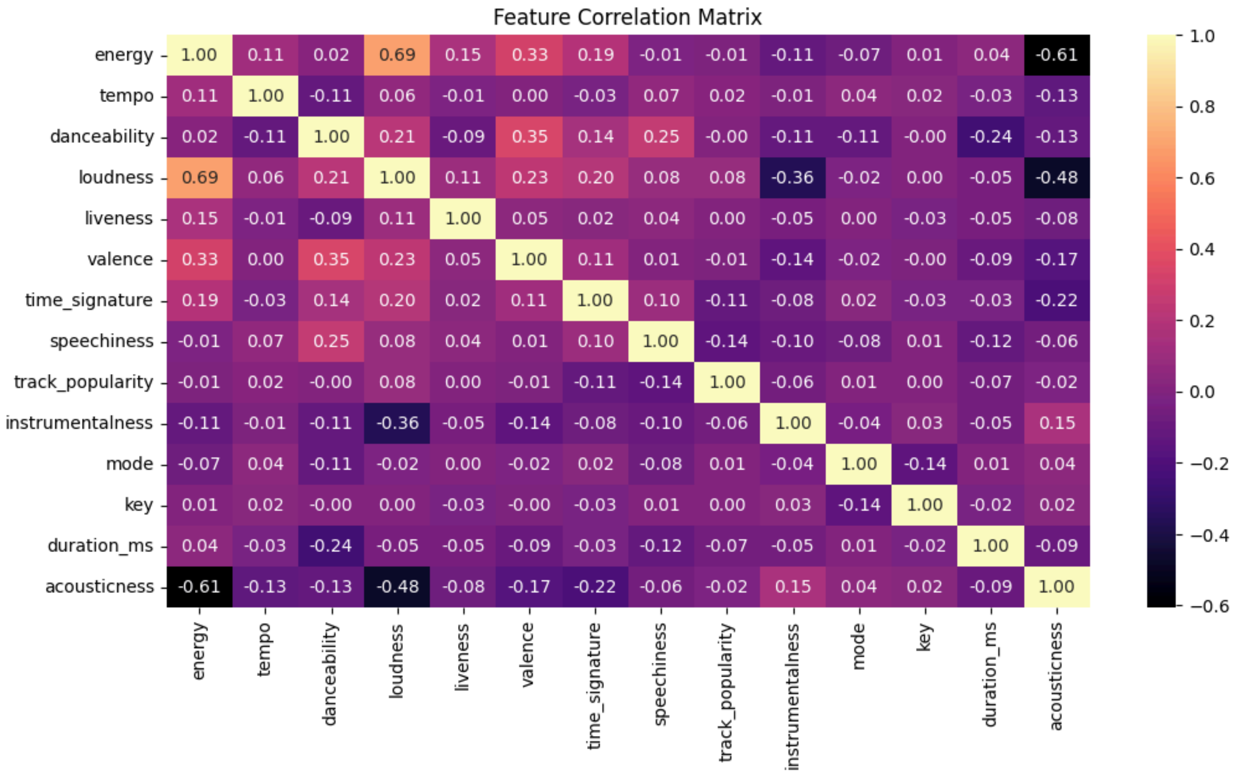
**Methodology & Data Collection:**

The data that I am using for the project was taken from a database on Kaggle, courtesy of Solomon Ameh. According to the description, the data was collected via Spotify's APIs to collect different data about some of the most popular songs. I've linked the original database here.

## Objective 1: Identify what Factors Most to Making a 'Hit' Song

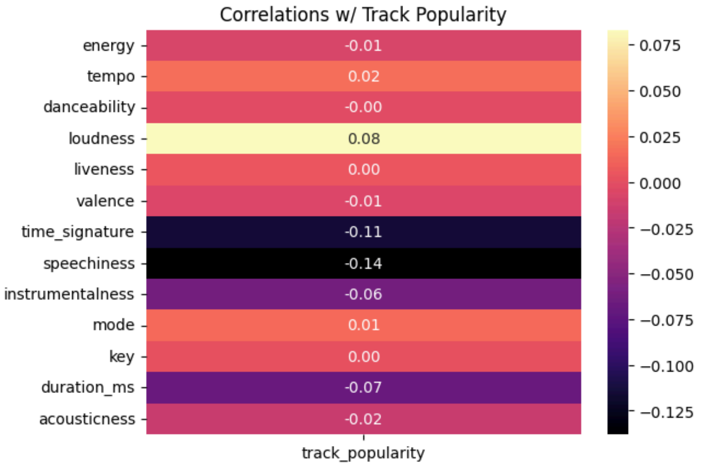
### Exploratory Data Analysis (EDA)

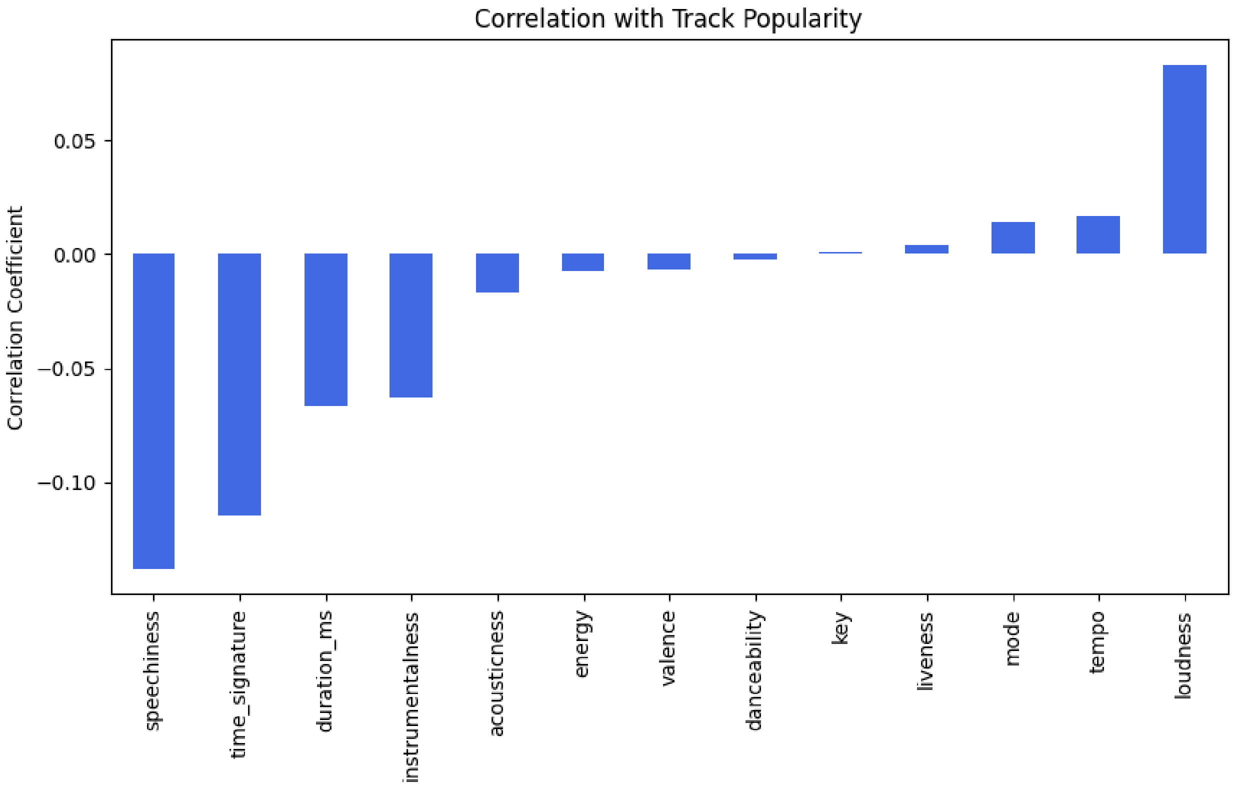
To kick off the project, I wanted to do some initial data analysis and get a good understanding of how certain variables correlated with others. I figured the best way to go about this was to create a heatmap and see all of the correlations between quantitative variables, shown below.



As we can see, there are not a lot of variables that correlate heavily with each other. The only somewhat strong correlation that I can see is that loudness correlates with energy, which is relatively self-explanatory. Weaker correlations include danceability and valence, along with valence and energy, which again seems relatively self-explanatory as valence is defined as 'musical positivity' conveyed by a track.

From this massive heatmap, I wanted to then hone in specifically on one column, the track popularity, addressing one of our key questions, as that would hopefully help us get a better understanding of what factors correlate with popularity.



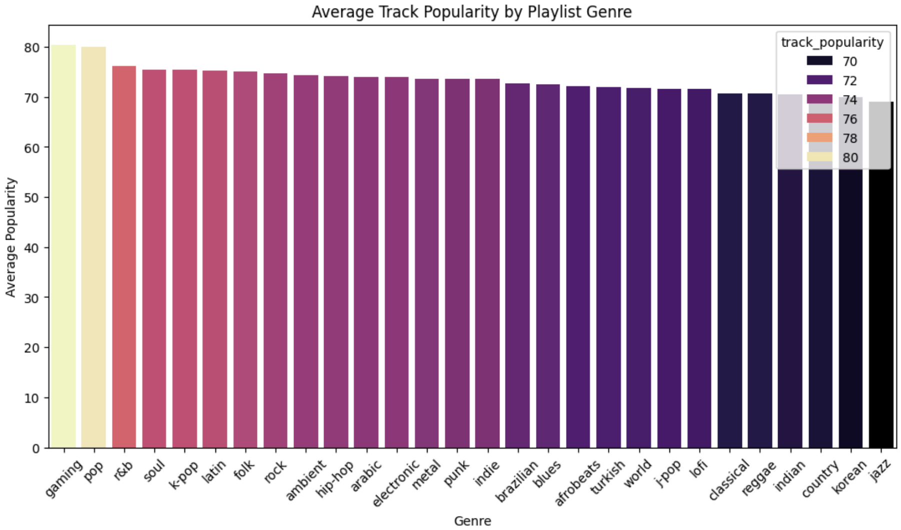


From these results, we see the:

1. The largest correlation with track popularity today is 'loudness'
    a. For me, this actually makes a good amount of sense since most songs that tend to be pretty popular are ‘poppy’ and upbeat, and therefore louder than more somber songs.
2. The largest Inverse Correlation with popularity is 'speechless', therefore, less wordy songs seem to actually be more popular
    a. This was relatively shocking to see since a lot of today’s most popular songs are hip-hop and rap, which tend to be more wordy. Especially during the early 2000s, when artists like Eminem were extremely popular, it would seem contradictory to say that speechiness has an inverse relationship with popularity. It could be possible that ‘speechy’ rapping styles have fallen out of popularity relative to more melodic flows like that of Drake/Kendrick Lamar.
3. Additionally, smaller factors like duration and time signature play a role in inversely correlating with the most popular songs on Spotify currently
    a. These make a lot of sense as well since most ‘pop’ hits that we see are typically around 2-4 minutes long. Any longer than that, and it becomes repetitive, and any shorter tends not to be popular either, as it’s too short. Later on in this project, I built out a polynomial graph that plotted popularity with track duration, and it seems like the

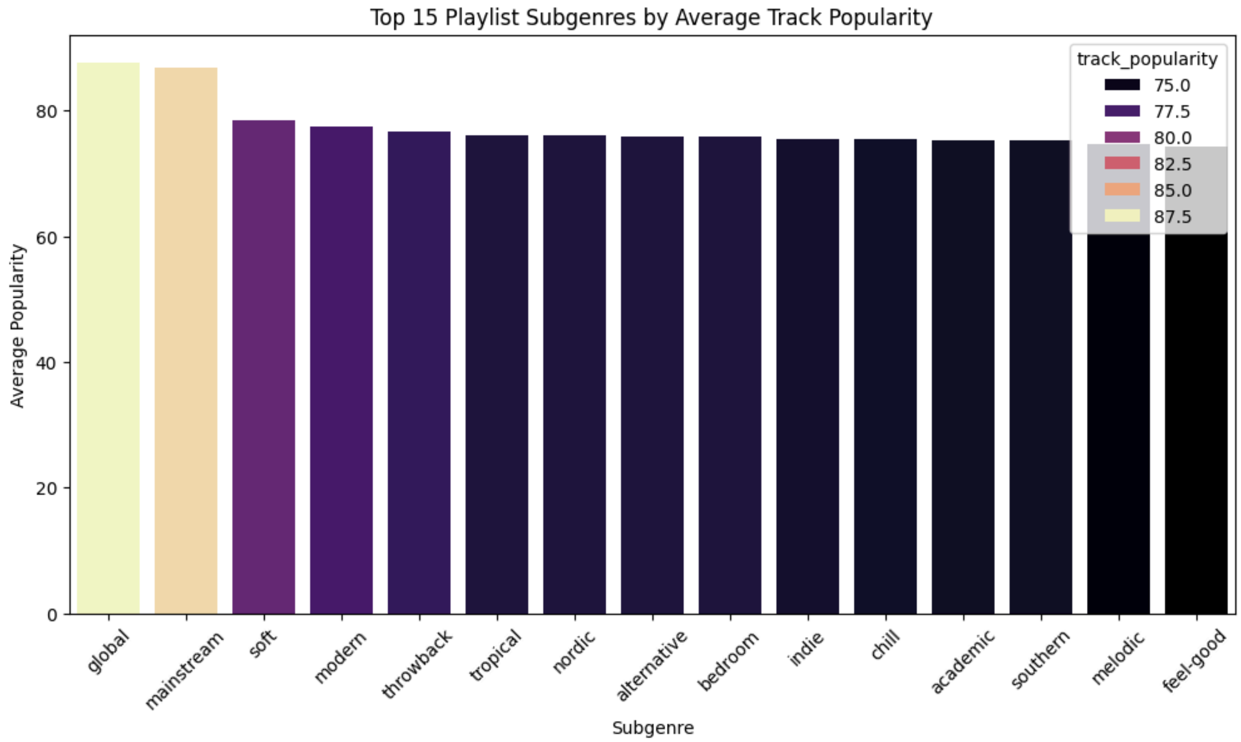
most popular ones are within the 150 to 200-second length, which aligns well with my prediction from earlier.

Following up, I wanted to then move toward the categorical variables to see if there were any major correlations between factors like genre/artist with popularity. Starting with genre, I plotted average track popularity by the different genres:



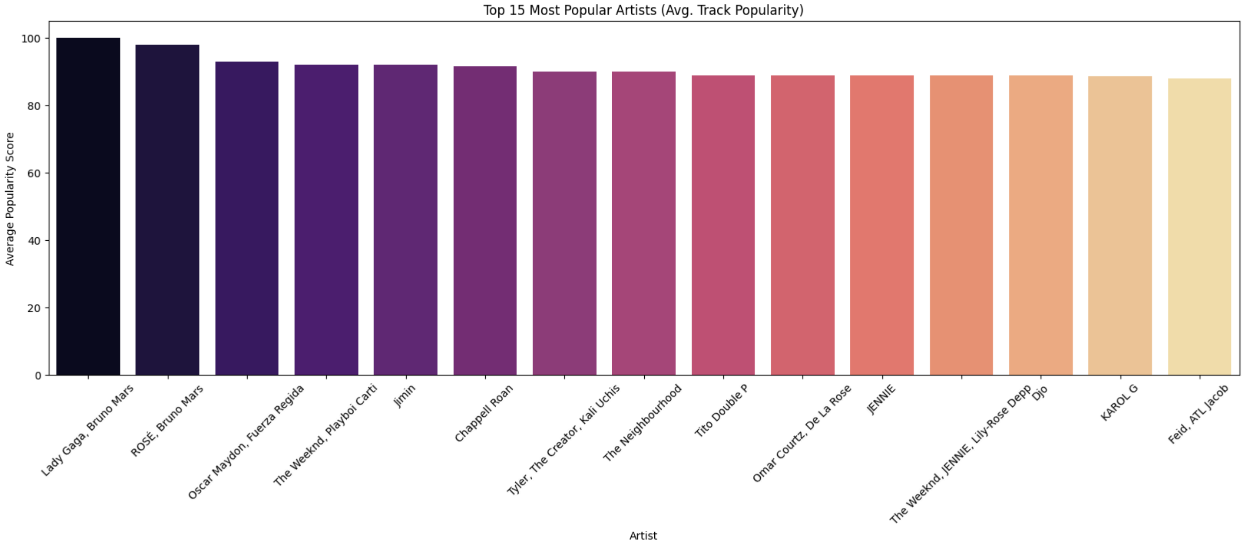
I'd say that most of the results make sense with what I expected, with pop and R&B being at the top, however, I was definitely not expecting 'gaming' to be close to the top, let alone taking the number one spot. I'm still a little confused by this result, but I assume that it's up there because of how niche it is, and therefore has a very small but strong follower base which could increase the average popularity of songs in that category.

Because I wasn't satisfied with those results, I wanted to take a deeper look into subgenres to see which ones were most correlated with popularity:

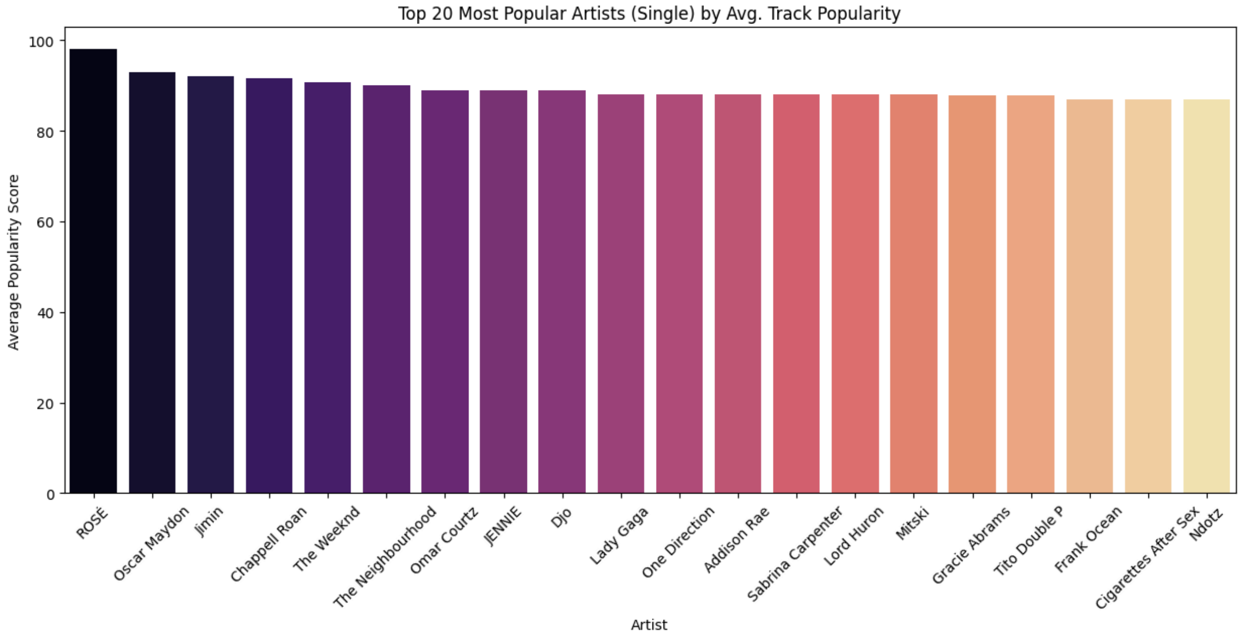


Taking the top 15 most popular subgenres, I'd say that the top 15 are definitely in line with what I expected, with global, mainstream, soft, and modern all being in the top 5.

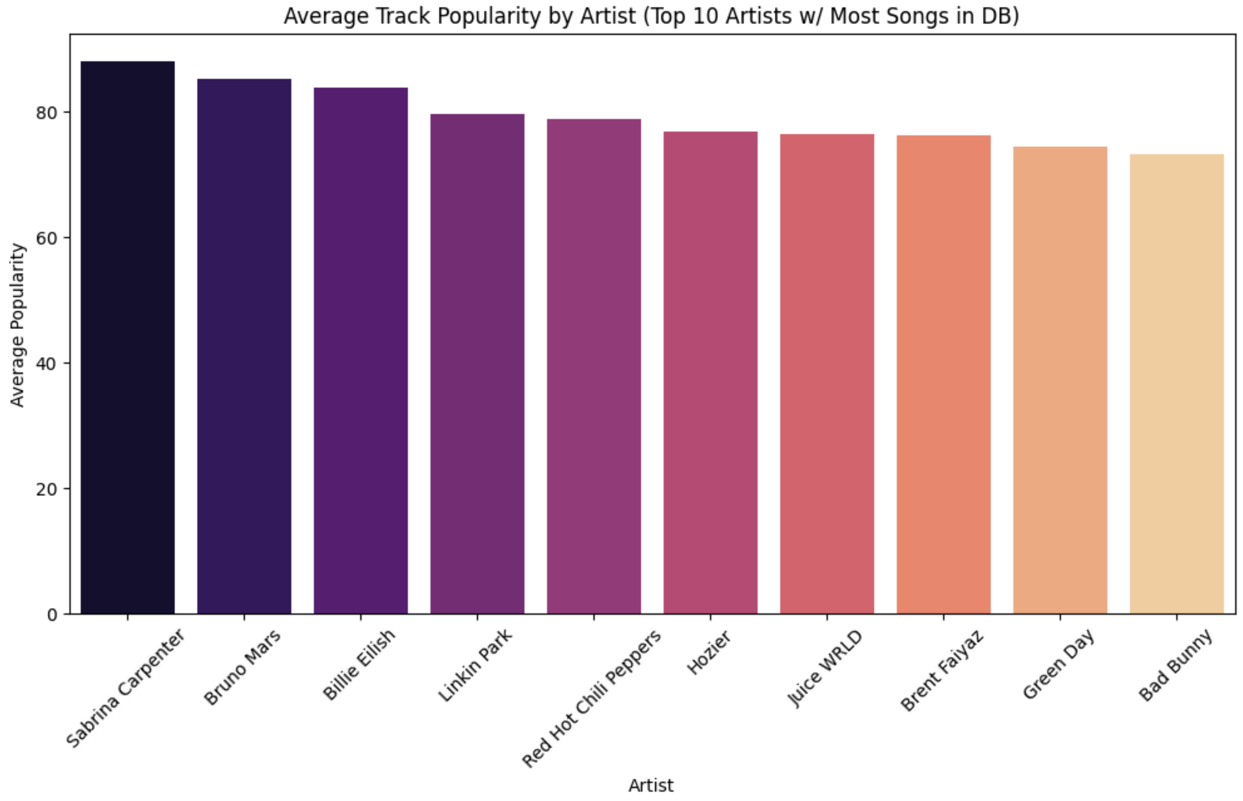
Secondly, I took a look at the top 15 artists that had the highest average popularity score



However, from these results, we see that it includes multiple artists in some columns, which is definitely not something that we are looking for. I hypothesize that it is doing this because there are a handful of super popular tracks that have certain combos of artists (ie, the top result being Lady Gaga & Bruno Mars's "Die with a Smile").



Lastly, I wanted to evaluate whether artists who created more music also correlated with the top-performing artists in the dataset. Below, I have listed the top 10 artists by song count in the database.

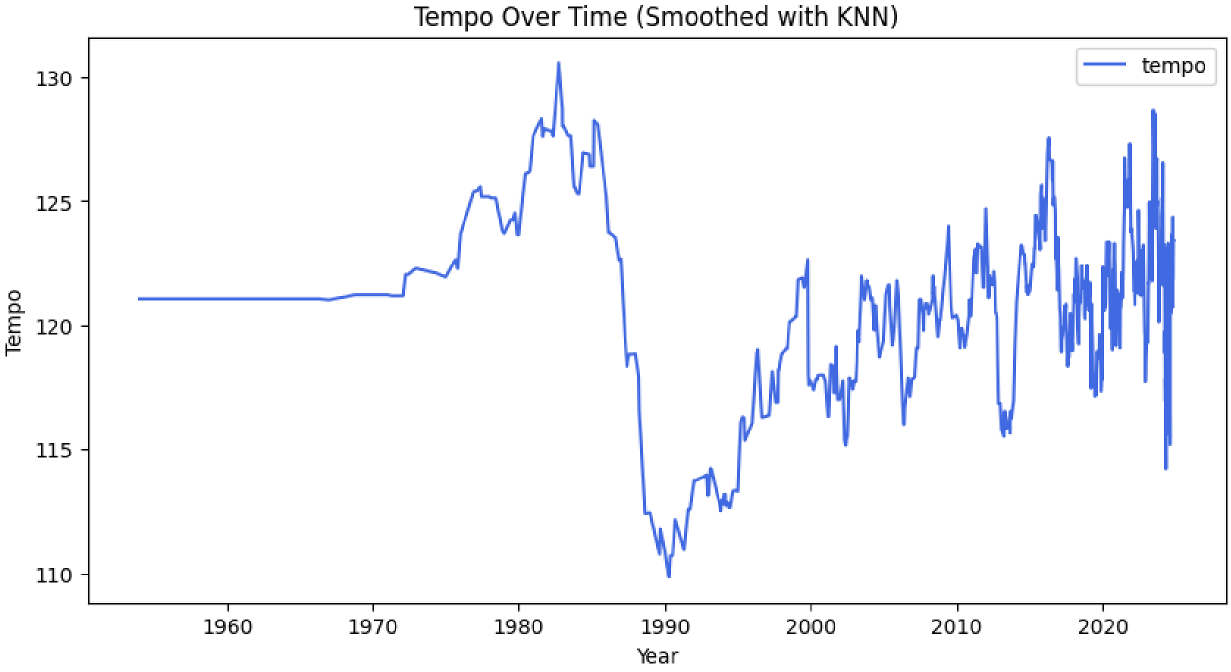


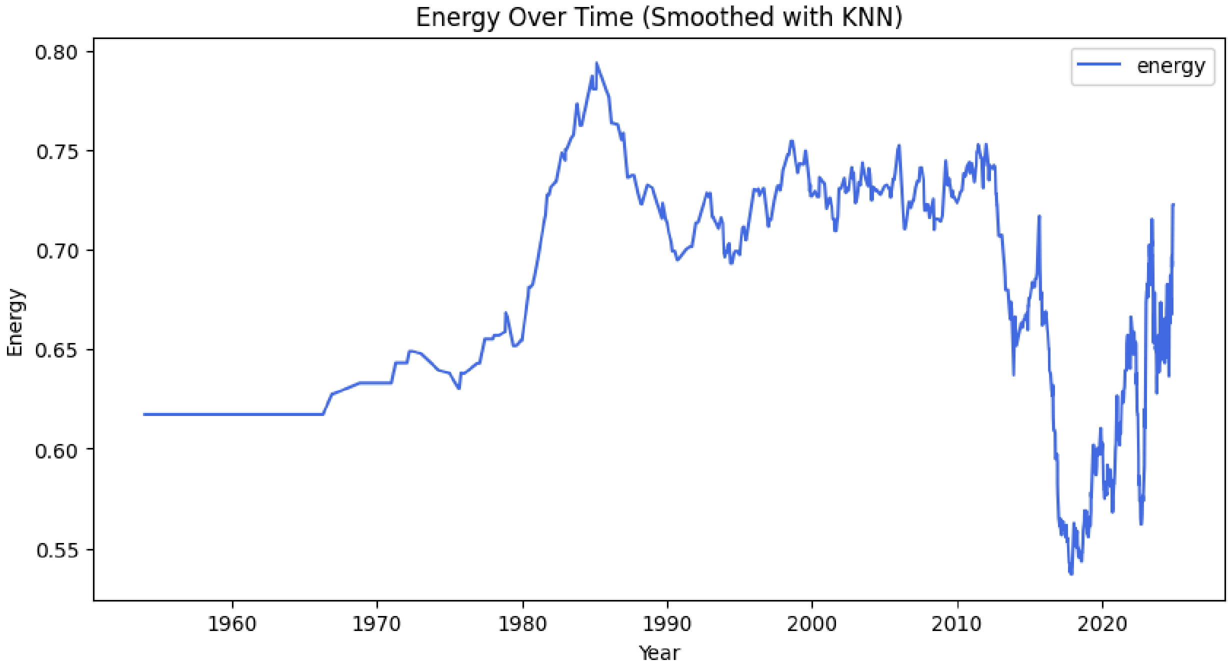
There is only minimal overlap between the two, with just Sabrina Carpenter being on both. Similar to the hypothesis from earlier, this makes sense because as there are more songs (test points), the average tends to converge to its true value.

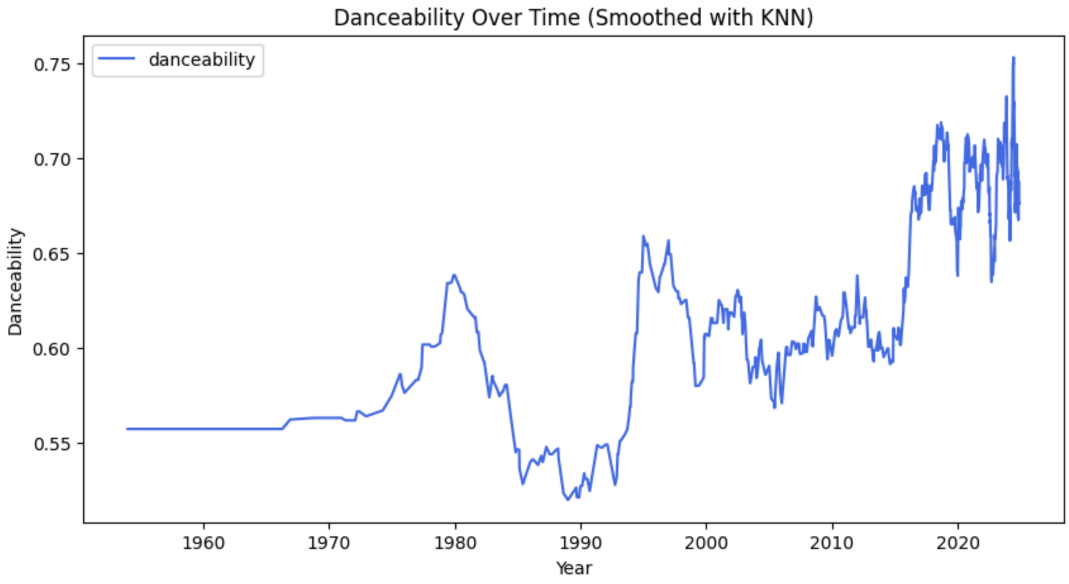
## Objective 2:Evaluate the Evolution of Popular Music

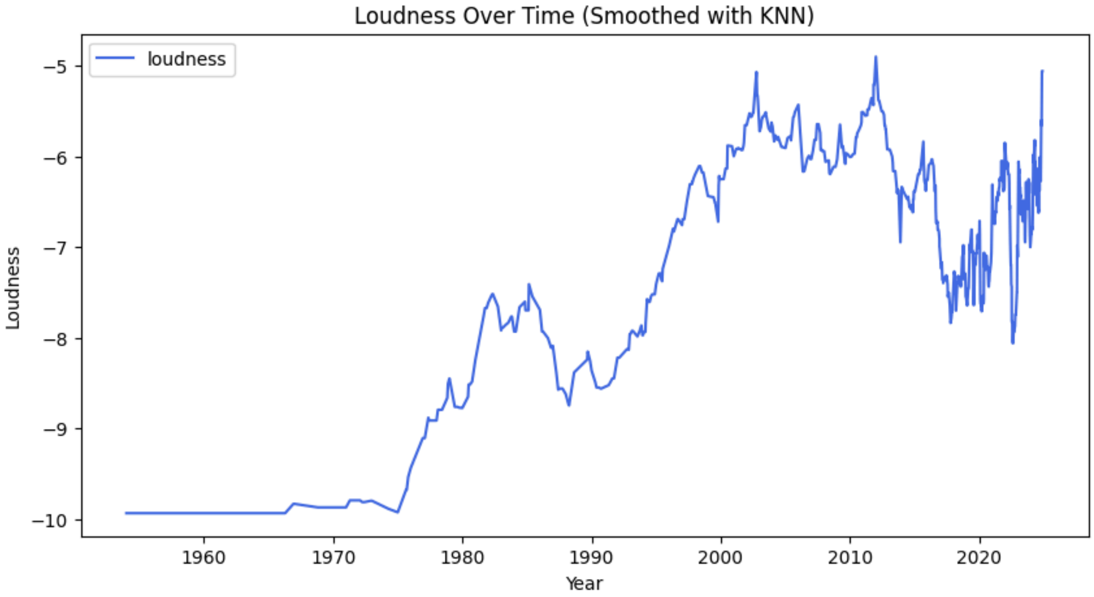
### Time Series Analysis

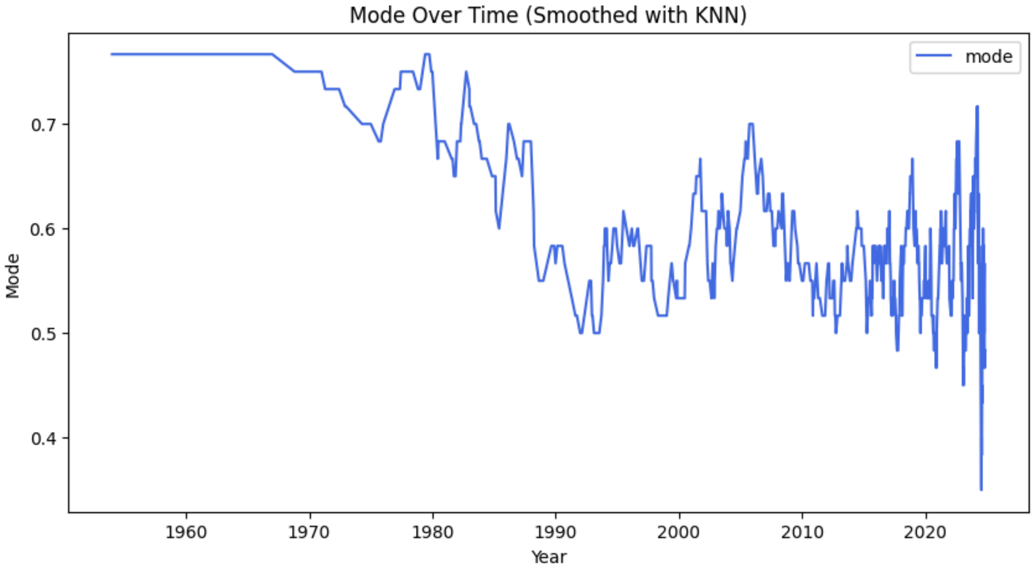
For this section, I created a series of time graphs that tracked how various characteristics of songs evolved over time. For this experiment, I chose 5 quantitative factors: Tempo, Energy, Danceability, Loudness, and Mode. To create this graph, I used a KNN model to smooth out the lines, using 60 neighbors, and tracked how it changed based on the release date. I found that 60 neighbors was the best number after playing around a little with the graphs, as any less would look too volatile, and any more seemed unnaturally smoothed. The resulting 5 graphs are all pasted below.


Tempo Over Time (Smoothed with KNN)
tempo
Tempo
130
125
120
115
110
1960
1970
1980
1990
2000
2010
2020
Year


Energy Over Time (Smoothed with KNN)
energy
Energy
0.80
0.75
0.70
0.65
0.60
0.55
1960
1970
1980
1990
2000
2010
2020
Year


Danceability Over Time (Smoothed with KNN)
danceability
Danceability
0.75
0.70
0.65
0.60
0.55
1960
1970
1980
1990
2000
2010
2020
Year


Loudness Over Time (Smoothed with KNN)
loudness
Loudness
−5
−6
−7
−8
−9
−10
1960
1970
1980
1990
2000
2010
2020
Year


Mode Over Time (Smoothed with KNN)
mode
Mode
0.7
0.6
0.5
0.4
1960
1970
1980
1990
2000
2010
2020
Year

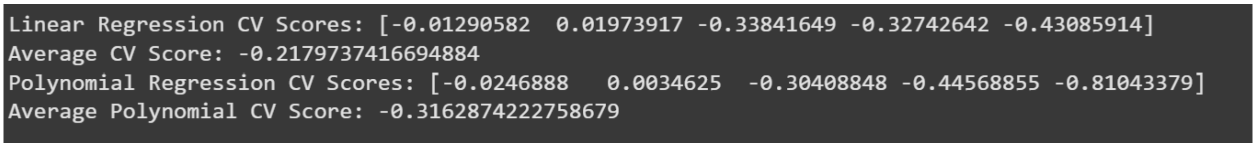
**Results from trend analysis:**

1. Tempo in popular music dropped significantly in the 80s but has made a steady rise ever since the early 90s to the modern day
2. Energy in top songs took a major drop in the 2010s, however, it again made a large comeback in the late 10s/early 20s
3. Since the start of the 90s, danceability has been an increasingly large indicator of popular music
4. Likewise, we see that loudness has also increased a lot over time, with major increases in the 90s decade, but dropped a little during the 2010s, with a massive comeback in the 20s.
5. Lastly, mode (I believe 1 meaning major and 0 meaning minor) has seen a massive shift towards minor scales in the last few decades.

## Objective 3: Statistical Analysis on the Dataset

### Linear & Polynomial Regression Models

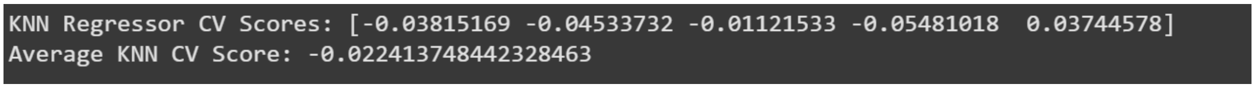
I attempted to use 2 predictive models to explain track popularity: linear & polynomial regression.



```
Linear Regression CV Scores: [-0.01290582  0.01973917 -0.33841649 -0.32742642 -0.43085914]
Average CV Score: -0.2179737416694884
Polynomial Regression CV Scores: [-0.0246888   0.0034625  -0.30408848 -0.44568855 -0.81043379]
Average Polynomial CV Score: -0.3162874222758679
```

Clearly, from the results, both performed poorly, with linear regression having an average cross-validation score of -0.218. The negative score indicates to us that the model did worse than simply predicting the average popularity for all songs. Ultimately, this linear regression model suggests that the relationship between quantitative features and popularity is relatively weak/nonlinear. However, I then proceeded to make a polynomial model, which did even worse than the linear one, with an average polynomial CV score of -0.316. I assume that the model here overfitted the training data and therefore failed to generalize well with new data. As a whole, it seems that there is no polynomial relationship that song popularity follows, which I believe to actually be good for the music industry as a whole, as there isn't a defined "recipe" to follow to create a hit song.

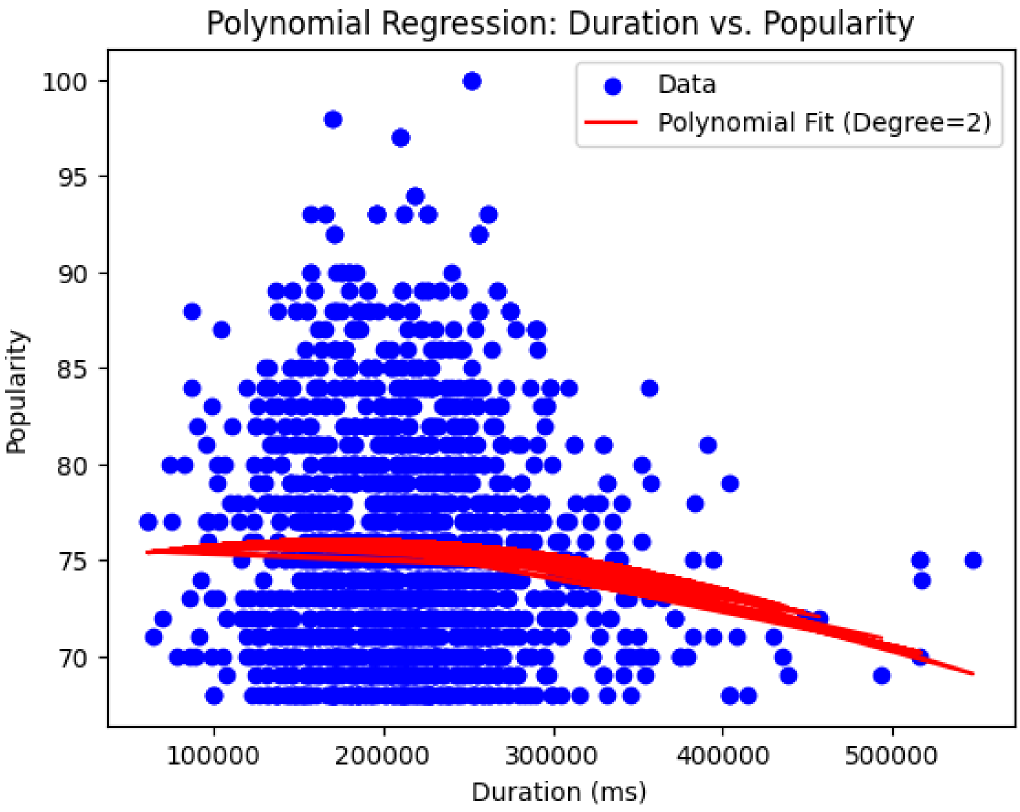
Additionally, I conducted a KNN Regressor, which smoothed some factor trends, but it still didn't perform that well since the average CV score was still negative. Overall, none of these models proved to be very effective at predicting popularity.

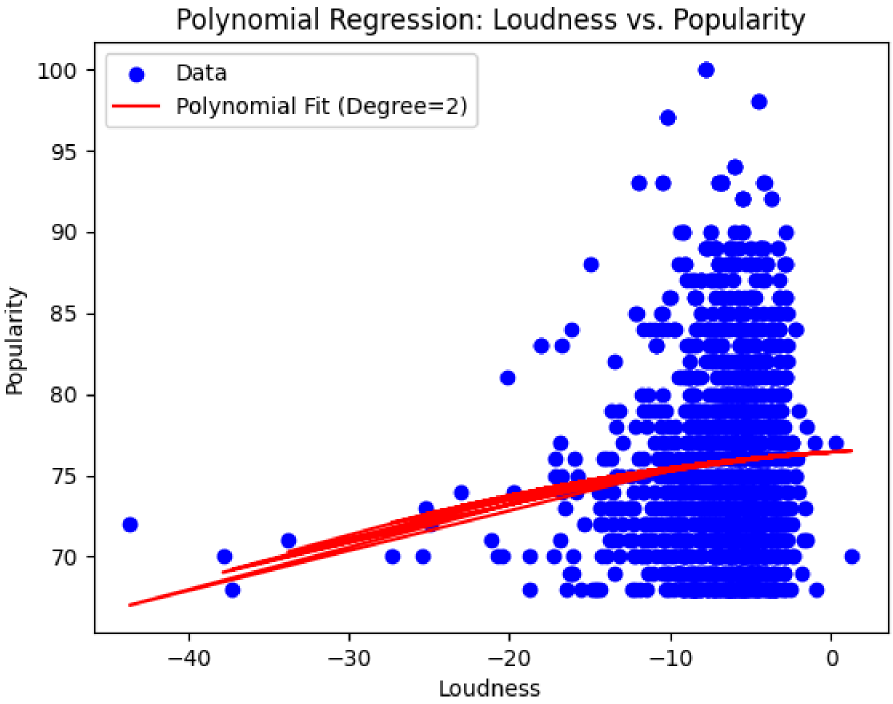


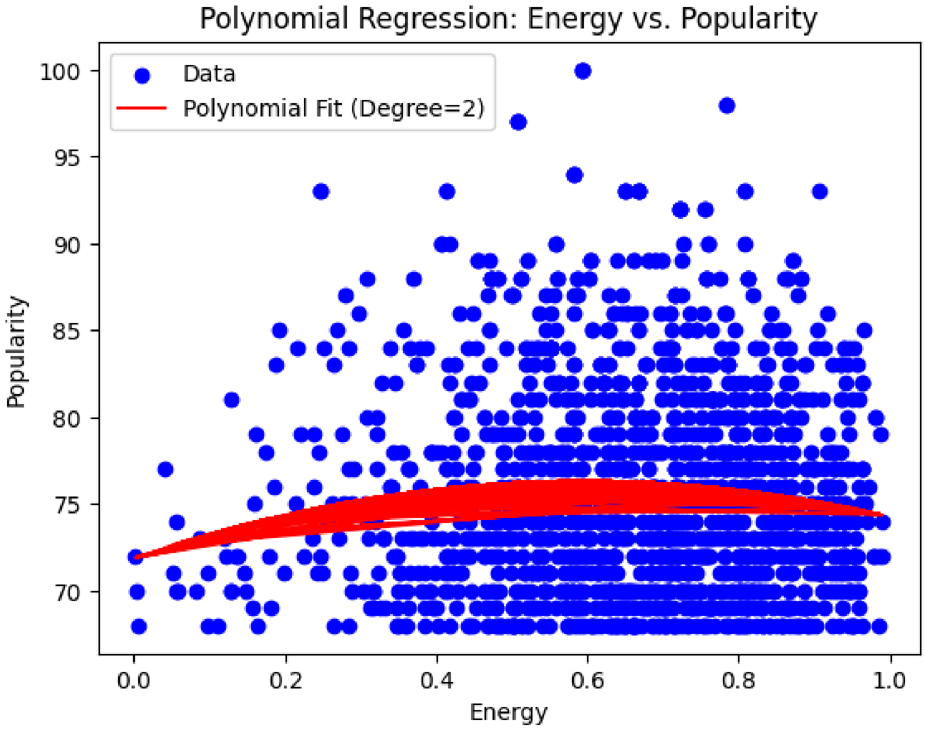
```
KNN Regressor CV Scores: [-0.03815169 -0.04533732 -0.01121533 -0.05481018  0.03744578]
Average KNN CV Score: -0.022413748442328463
```

Lastly, I conducted some statistical tests to see whether there was a statistically significant factor that differed between normally popular and the most popular songs. I split the dataset into two, popular and normally popular, based on whether a song's popularity was greater/less than the median. From there, I iterated through the various factors and saw which factors had a statistically significant difference between the most/normal popular songs. In the end, we only saw that 'duration'  had a somewhat noticeable effect, with a t-stat of -0.95 (less than one standard deviation away from the mean) and a p-value of 0.34 (which is above our threshold of 0.05).  Ultimately, it seems like there is no sole predictor of a song's success.

Lastly, I wanted to do some additional plotting of factors with popularity. The three I thought were most interesting were duration, loudness, and energy. Below are the polynomial fit trends.



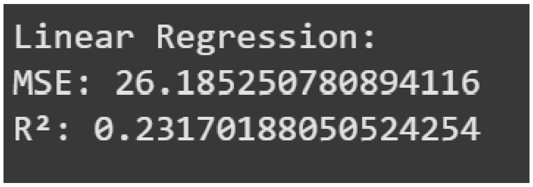




## Objective 4: Create Predictive Models

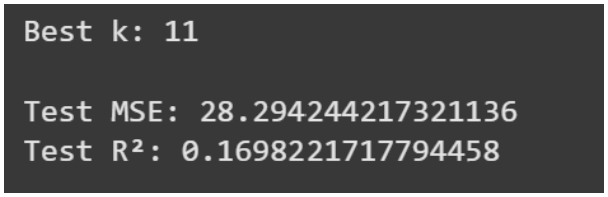
### Linear & KNN Models

Linear Model Results:



```
Linear Regression:
MSE: 26.185250780894116
R²: 0.23170188050524254
```

KNN Model Results:



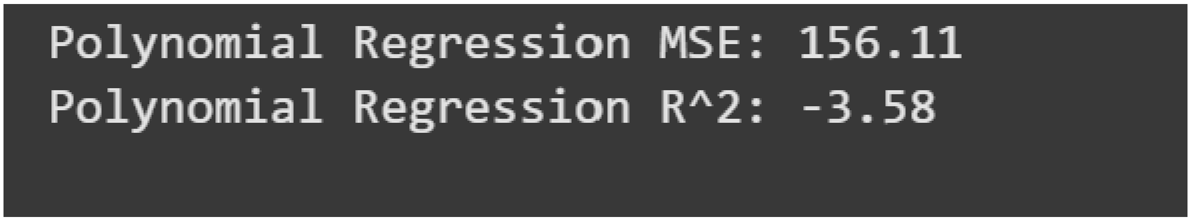
```
Best k: 11

Test MSE: 28.294244217321136
Test R²: 0.1698221717794458
```

As we can see from the results, neither is ideal, as both R^2 values are within the 0.1-0.25 region, which tells us that our model only captures at most 25% of the variance in popularity scores.

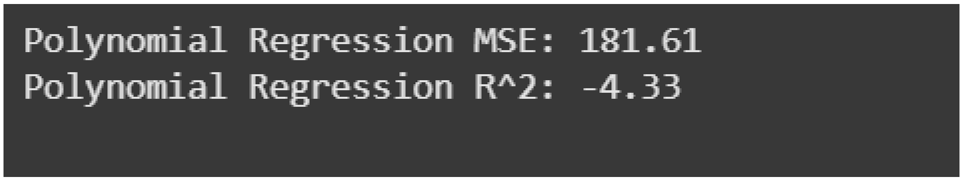
However, between the two models, I was a little surprised to see that the basic linear regression model actually outperformed the KNN model (even after optimizing for the best number of k neighbors via GridSearch).

Ultimately, because both models were less than ideal, I decided to try adding more complexity to these models. With the linear regressor, I tried to incorporate polynomial features, but it ended up making the model even worse, resulting in a negative R^2:



```
Polynomial Regression MSE: 156.11
Polynomial Regression R^2: -3.58
```

To combat this, I tried using the ridge regressor instead of the linear one, but it still undercompeted relative to the previous models.



```
Polynomial Regression MSE: 181.61
Polynomial Regression R^2: -4.33
```

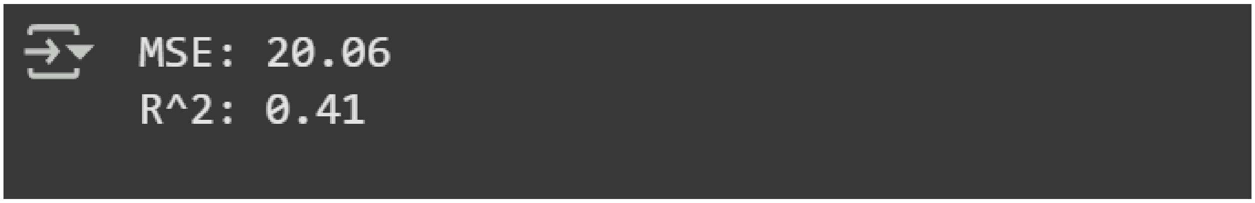
This time around, with the Ridge regressor instead of the Linear regressor, the MSE dropped significantly from 181 to ~35, almost a 4.5x difference.

However, the R^2 value is still negative with both models, indicating that the model's predictions are worse than simply using the mean of the dependent variable as a prediction.

Ultimately, since both of these attempts were still unsuccessful, I elected to try more complex models like the Random Forest and Decision Tree models to see if they were any better at predicting the popularity of songs.

**Random Forest Regression Model:**

Results:

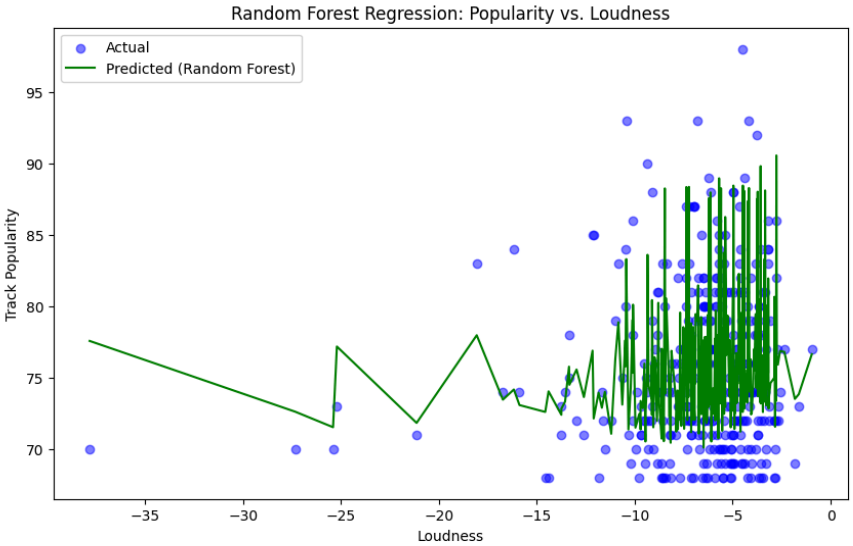


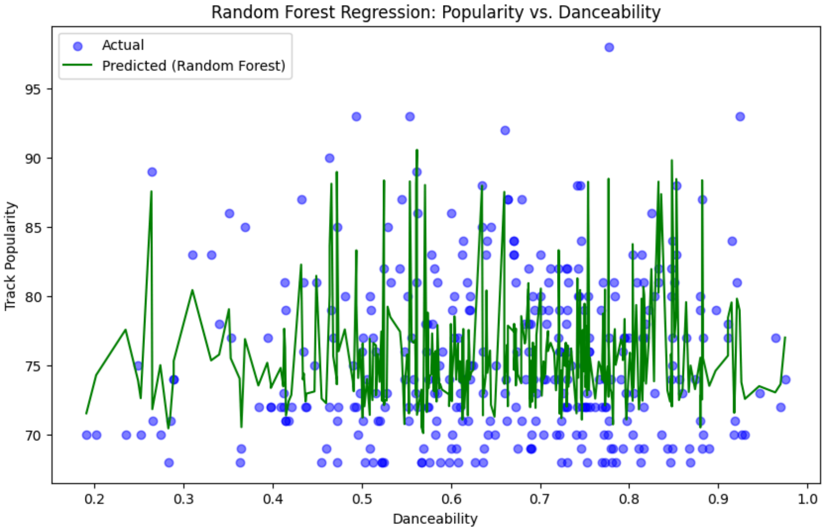
```
MSE: 20.06
R^2: 0.41
```

Compared to the 2 previous models we tried (simple regression and KNN), this one did significantly better as the R^2 improved by almost double that of the simple regression and more than double the KNN. Additionally, this MSE is the smallest as well, which is a great sign.

In terms of why I think this model performed better, I think that it succeeded because I'd doubt that music trends would be linear. I wouldn't expect there to be a perfect linear relationship between popularity and tempo or energy (as there should be a sweet spot before it begins performing worse). Therefore, it makes sense why this dataset demonstrates nonlinear patterns, something that a simple linear regression clearly performed a little more poorly against. Additionally, a model like random forest is slightly better at handling high-dimensional categorical data (like genre), without potentially overfitting like a KNN model might.

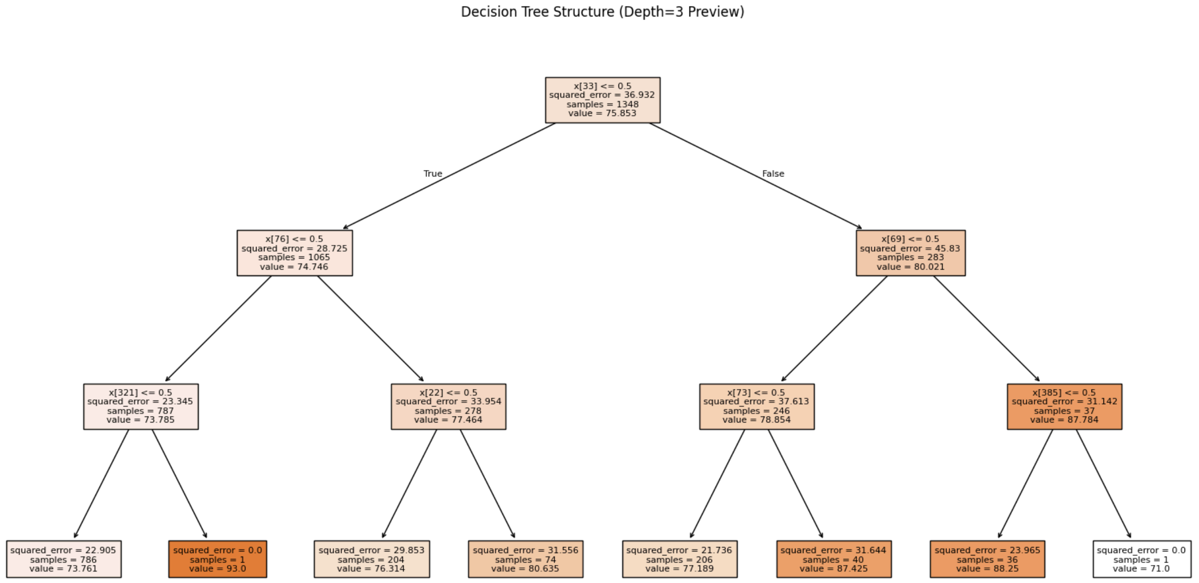
Additionally, I've also attached two visuals below showing the random forest regression model at work. One time, with the 'loudness' trait which we deduced from the EDA was a significant predictor, and also with danceability.

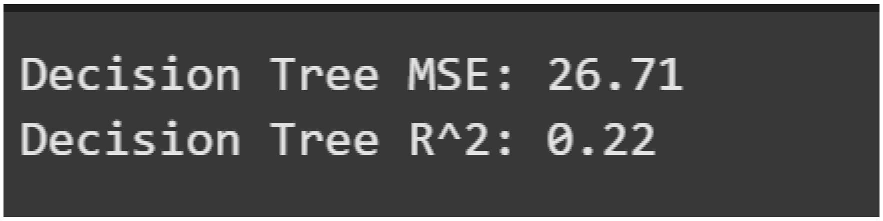




## Random Forest Regression Model:

Lastly, I elected to try building a Random Forest model to get a better understanding of how the model actually arrived at its prediction, and also to visualize the decision paths as shown below.

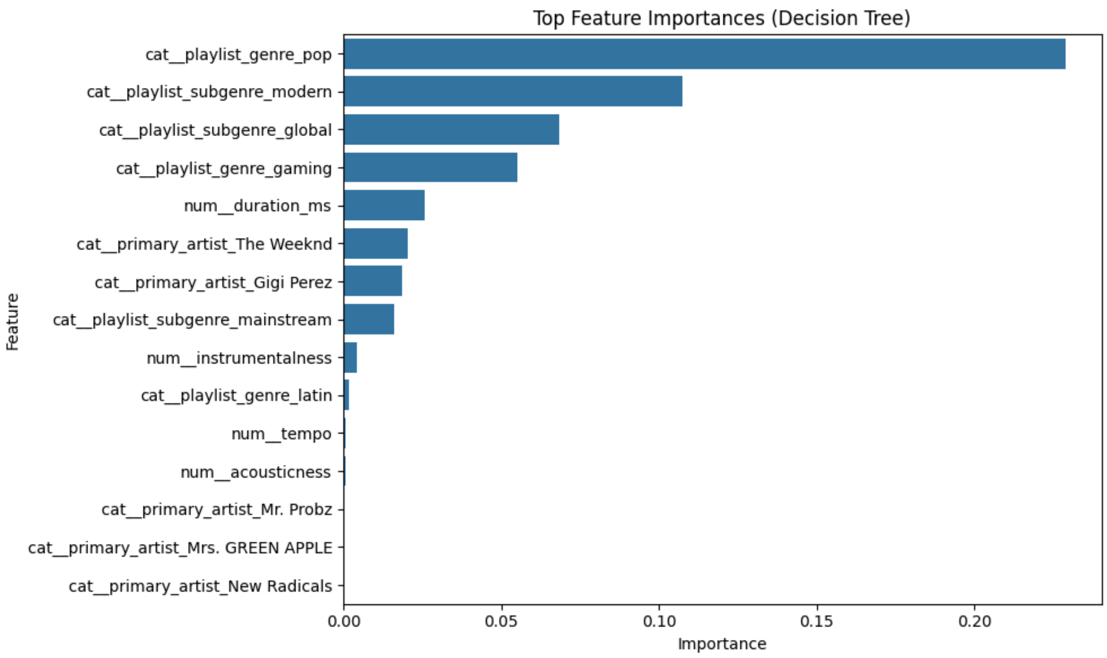




```
Decision Tree MSE: 26.71
Decision Tree R^2: 0.22
```

However, in terms of results, I noticed that the model did worse than the simple regression and the random forest model. Because decision tree models are more susceptible to overfitting data, I assume that it did in this case, and thus performed worse. It especially makes sense in a database like this music database, where there are clearly less visible trends that contribute to consistently creating a 'hit' song. Even in the EDA phase, we saw that there were a few trends, and thus, this model likely captured too much of the 'noise' in the dataset.

However, this decision tree was good in telling us what characteristics of a song it prioritized most, as shown below from the screenshot.



The model's weighting in 'top features' actually makes a good amount of sense as it identified some of the most correlated features with popularity that we discovered during our EDA phase, like genre.

Also, it put a significant weighting on duration, which we also discovered during EDA was a major quantitative determinant of a song's popularity. Lastly, it's interesting to see that the model identified some specific artists like The Weeknd, who consistently make popular/hit songs.

Ultimately, comparing all 4 models together, it's clear that the Random Forest one did the best job, but still imperfect as the R^2 only reaches 0.4, indicating that it can only predict 40% of the variance in popularity scores.

## Objective 5: Reflections, Next Steps, and Improvements

**Main takeaways:**

I found that certain complex models fared better than simpler models (ie, in the case of the random forests model performing better than KNN and regression in terms of R^2 and MSE). However, there were also cases when more complicated models simply did worse than their simpler counterparts. In the case of the linear model that factored in polynomial features, it received a negative R^2 and a significantly worse MSE and R^2 than a simple linear regression.

Another key takeaway is that none of the models did particularly well, as the highest R^2 was still only in the 0.4-0.5 range. The moderate R^2 values tell us that while the model might have captured some structure, music popularity is still extremely difficult to predict and it's extremely multifaceted (and maybe even has some luck/randomness involved). This would make sense as there are confounding factors like social and cultural factors, that this model simply couldn't account for.

**Next Steps:**

In terms of next steps, we could definitely have more features involved to make the model more holistic. For example, like I mentioned earlier with the social factors, maybe including external data such as YouTube or TikTok trends, playlist followers, or artist social media. In recent years, we've seen that many popular songs are also those that go extremely viral on TikTok, so I would not be surprised if there was a major correlation there.

Secondly, we could include temporal elements like release year that would reflect how music tastes change over time.

Overall, I believe that this was a good first step in predicting music popularity, but nowhere close to being useful, as our models' R^2 values are simply not good enough. In the future, I hope to incorporate some additional features/refine current models to make it more exhaustive and accurate.