# Analyzing Spotify Data

Lucas Gaspar
*Bachelors in Cyber Security*
*Wentworth Institute of Technology*
Boston, Massachusetts
gasparl@wit.edu

***Abstract*— In the day of digital music streaming, Spotify holds an impressive grip on the market. This makes it important for artists to know what it is that drives plays on Spotify, as well as how songs are classified for recommendations. I will be analyzing the features of songs pulled from the Spotify API to predict how popular they may be based on these features. I also will predict what genre a song belongs to which can be used for song recommendations. Finally, I will create visualizations that give insight into trends in songs throughout history to better learn from these mistakes or successes. Using this information, we can create a better music platform, as well as improve the music that gets uploaded to it.**

***Keywords*— *Spotify, Music, Machine Learning, Genre, Marketing***

## I. Introduction (*Heading 1*)

Spotify is the world's largest music streaming platform, hosting countless artists ranging from international icons to independent artists trying to make a name for themselves. Each song uploaded to Spotify has its listening activity tracked, along with various characteristics of the song. By analyzing these traits, we learn about what makes a song reach the top of the charts as well as what might keep some from getting there. Additionally, genre analysis and prediction can be done, leading to more accurate Spotify recommendations and song grouping. This would result in an overall better experience for users as they can get songs recommended for them that are closer to what they listen to. Combining all these ideas, we can help artists with song creation by knowing what features users enjoy listening to.

Song analysis has been an important and well researched topic for as long as we have been able to access it easily. Music and its creation are not stagnant, though. This allows for meaningful conclusions to be drawn from nearly all instances of analysis. Analyzing different streaming platforms could produce different results or even just the time of the year the data was taken. Additionally, the common public opinion changes frequently, adding another layer of complexity to this topic. This is why it can always be valuable to perform analyses such as these.

For the genre analysis, I was very curious to see how effectively songs could be automatically assigned into genres based on their traits. The definition of certain genres seems to vary among people and overlap into each other. I expected this to lead to interesting conclusions and be a difficult subject but still provide good insight into how songs can be related to each other and organized.

## II. Datasets

### A. Source of dataset

The data that I will be analyzing is titled, "30000 Spotify Songs". It was uploaded on Kaggle in 2024 by Joakim Arvidsson. Although it is not officially published by Spotify, we can confirm its validity by the fact that all information contained within the Spotify API can be accessed and used for free. The dataset's creator collected this information using python scripts designed for this purpose. Cross referencing this dataset with others created using this script led to consistent results as well, guaranteeing this is true Spotify data.

### B. Character of the datasets

The dataset contains over 32,000 entries, each representing a specific song hosted on Spotify. Each song has stats about the specifications of the song such as its key and tempo, as well as the song's creators and its release date. The data ranges across all genres and varying levels of popularity with no bias to either. This should allow for accurate analysis of popularity as well as on songs with varying release dates. When it came to cleaning and reorganizing the data, there were a few things I needed to focus on. One of these was to make the upload date variable function as an incremental number. This allows me to observe trends over time. This new value was not stored as a new feature and was only calculated before the analysis that required it.

There were some changes that needed to be made when addressing specific questions. The track_popularity feature is a measure from Spotify themselves which is calculated from the number of plays a track has gotten in the past 30 days [2]. Due to the nature of the songs being collected randomly, there are many entries that have a popularity of 0, indicating that the song has had no recent plays. Rather than proving that a song is unpopular, these zero values simply indicated that there was no data for these songs yet. Because of this, I chose to omit these entries when looking to analyze popularity. I also created a list titled "features" that is used extensively in testing to quickly access the variables that I deem most useful. These features include: 'energy', 'tempo', 'danceability', 'loudness', 'liveness', 'valence', 'speechiness', 'instrumentalness', 'acousticness', and 'duration_ms'.

Additionally, I had to make some decisions regarding what

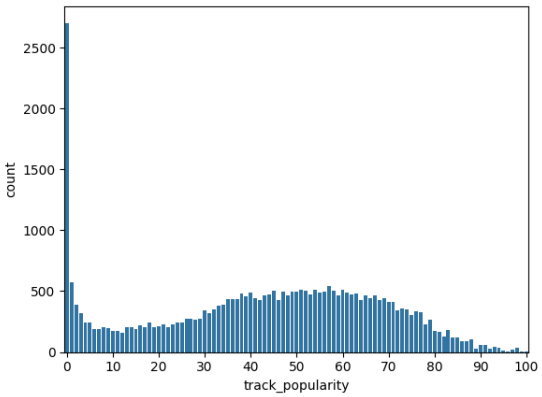


*Figure 1: Distribution of data popularity*

type I considered my data. When looking at the popularity value, it seems categorical due to the way that the data is

979-8-3315-0993-4/25/$31.00 ©2025 IEEE

distributed. Instead, I chose to treat it as numeric. This allowed for easier interpretation of my predicted values as well as compensating for the fewer number of entries for tracks at high or low popularity values.

| track_name | playlist_genre | energy | instrumentalness |
|---|---|---|---|
| Typhoon… | edm | 0.884 | 0.341000 |

(Example data entry)

## III. Methodology

To study this topic as accurately as possible, I implemented several different models and visualizations, each specializing in looking at the data in a new way. I achieved this spread of interpretation by using several categorical and numeric machine learning techniques to discover what best represented the data. I also included numerous diagrams and visualizations for easier analysis and interpretation of the data/results. The final models used are shown below.

### A. K Means Clustering

Before doing any supervised learning to make predictions in the data, I wanted to better understand it. I achieved this by performing K-Means Clustering to see if the data could be grouped in ways that would make interpreting it easier. I attempted to split the data into different numbers of groups and tracked the mean variables of each group. I made my conclusions by simply observing this output. I chose to implement this model first to become more familiar with my data's features and how they might be correlated.

### B. Random Forest

To tackle my classification problem, I implemented a random forest decision tree that uses the "features" list described earlier. This random forest model creates many decision trees with varying features and lengths to find the optimal tree. I split the data into testing and training data and used the RandomForestClassifier() method from the sklearn.ensemble package on the data designated for training. The result of this is then tested on the remaining data to check the model's accuracy. Splitting this data is not as important as with other models due to he nature of making many trees and testing them all but still provides further insight. This model does a great job of identifying important features and finding ther correct model size. A downside of this method when compared to normal decision trees is that there is no single tree to visualize and it is significantly more computationally expensive.

### C. Linear Regression

Another model used in my research was linear regression. This allowed me to predict a numeric value for my question regarding track popularity. This was created with the sklearn package to create the model as well as calculate the Mean Squared Error (MSE), a measurement of how accurate my model's predictions are. Ultimately, this method helped me draw conclusions on what features influence certain variables. There are still some glaring issues with this method such as how it does not guarantee a prediction within the possible popularity values.

### D. Visual Analysis

I used visualizations to demonstrate how some variables change over time. This allows for an easy analysis of trends and relationships in the data that we may not have anticipated. I used standard pandas and matplotlib packages to graph stats over time. This required me to convert the upload dates into numbers that can be used in the analysis. I plotted many different song features and found the most interesting results when looking at loudness. I grouped songs into one-year bags for analysis.

## IV. Results

After implementing my methods outlined above, I began analyzing my results. Here, we can see the capabilities of various machine learning and visualization algorithms in representing Spotify data. These discussions are all with the end goal of better understanding tracks to improve Spotify recommendations and help artists produce better songs.

### A. K Means Clustering Result

| | speechiness | instrumentalness | acousticness |
|---|---|---|---|
| 0 | 0.311875 | 0.010558 | 0.180496 |
| 1 | 0.072409 | 0.089509 | 0.508564 |
| 2 | 0.074502 | 0.014006 | 0.140182 |
| 3 | 0.071438 | 0.740474 | 0.071720 |
| 4 | 0.073663 | 0.024936 | 0.066593 |

*Figure 2: Example of clustered data showing instrumentalness & acousticness relationship*

When attempting clustering, it is important to find groups that split the data into easily interpretable clusters. These clusters can signify relationships in the data or traits that lead to entries being similar in the context of a target variable. In my testing, I did not uncover any highly grouped songs but did find some of their features that seemed to overlap with each other. Some trends I noticed are that songs with low instrumentalness also tended to have very low acousticness, but there was also another cluster where this was the exact opposite, where the acousticness was significantly higher than normal. I attempt to use this knowledge with understanding how genres can be assigned later.

### B. Genre Analysis Result

I used three different models in my analysis with the random forest version being the most effective by a significant amount. A 54% accuracy is a very competent result that vastly outperforms the null hypothesis. This proves the validity of this model. There are 6 different genres available in the playlist_genre variable with the most popular being EDM. If we create a null model and predict every song to be EDM we get an accuracy of 18.4%. This tells us that the random forest vastly outperforms the null model and has significant merit in its use. Predicting a song's genre in this way could effectively recommend songs to user or automatically sort them in playlists and libraries. It is also possible that even if a prediction is wrong, the predicted genre may be closer to the correct genre, allowing for loose organization to be made.

1) *Random forest Accuracy: 54.2%*
2) *Linear SVM Accuracy: 24.1%*
3) *KNN Accuracy: 27.9%*
4) *Null Model Accuracy: 18.4%*

### C. Linear Regression Result

My linear regression model allowed for some decent prediction of the data. Using the model with popularity values at 0 excluded, I had an MSE of 469. This came out to be a 6.36% improvement over the null model. This indicates that there may be significantly more at play when determining the popularity of a song. If we take the square root of these values, we can find about how much the average prediction was incorrect. In this case, the best model was off by 21.6 points in popularity on average. Note that since I am using linear regression and treating popularity as a quantitative variable rather than categorical, it is possible for the predicted value to be negative or a decimal answer, despite the track_popularity variable ranging from 0-100 in practice.

1) *Null Model MSE: 617*
2) *Filtered Null Model MSE: 501*
3) *Predicted MSE: 572*
4) *Filtered Predicted MSE: 469*

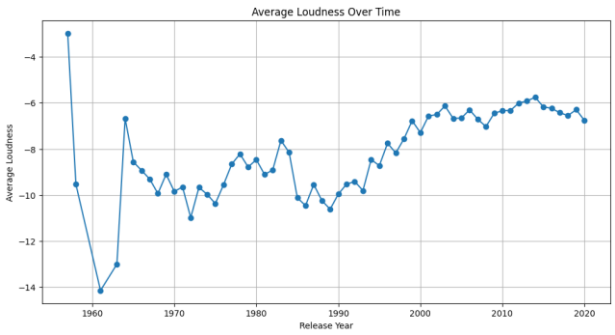


*Figure 3: Average Loudness Over Time*

### D. Visual Analysis and Modeling

One of the most interesting conclusions I was able to draw was using simple visual analysis. This plot, representing the average loudness of songs each year, shows that this value has been increasing over time. This may be due to new music mediums allowing for different levels of volume to be used, such as the introduction of the CD or the transition to music streaming services such as Spotify. The large spikes early in the data most likely represent areas where we have not enough data points or perhaps are a result of very old technologies not allowing for precise mixing of tracks. This is a known phenomena called the "Loudness War" and is harming the quality of music. It is important for artists and producers to not mix their tracks in a way that increases or is too far below the current average loudness to ensure normality and prevent further loudness creep.

## V. Discussion

Although my results carry a great deal of information regarding the patterns set by artists and users on Spotify, I also ran into many issues that held my research back. For example, the overall idea of what makes a song popular is more nuanced than simply the sum of its features. This is shown in the large error in my linear regression prediction of track_popularity. An artist already being popular or advertising on other sites outside of Spotify can contribute greatly to the number of plays a song will get.

Due to the uncontrolled and vast sea of music that exists on Spotify, pulling data randomly and hoping for discrete clusters to form is unrealistic. I believe this to be the case no matter what method is used due to the elasticity of genres, but I do believe that more could be done in this unsupervised learning approach. In the future, I would want to look at using different forms of distance such Gower distance to calculate these clusters. Another option that I could pursue is the implementation of feature reducing algorithms such as FAMD. This would work on my data even though my features are mixed and provide a new way to visualize my data in fewer dimensions.

When it comes to genre analysis, I am quite satisfied with the results that I have achieved. Since my predictions proved useful, I would like to expand this analysis to a larger list of genres and songs. Additionally, I am confident that I could form even stronger predictions if I included more variables such as a song's key or mode. This would require more work to be used in algorithms as these are categorical variables. It is for this reason that I decided to exclude them from all my testing. Looking at the complexity of the data, I feel as though neural networks might perform best in this type of environment.

## VI. Conclusion

Looking at the key takeaways from my data, many conclusions regarding how song recommendations on Spotify can be done can be made. This information will also provide insight to artists looking to discover what it is that makes a hit song. I have shown, to a certain extent, that predicting a songs popularity ranking can be done. This can be used to show users songs that have the potential to become more popular early. There is still a vast amount of error in this method though, indicating that there are many more variables that must be considered when looking to predict this. This becomes complicated when acknowledging that many of these factors do not exist in Spotify itself.

Genre prediction can also lead to better recommendations as users are more likely to enjoy songs related to what they already like than something completely different. Using only 10 features of the songs in my data, I was able to predict a song's genre with over 55% accuracy. This number has the potential to increase greatly with the use of more variables or different machine learning techniques.

A final piece of information to note is the surprising relationship between a songs loudness and when it was created. My data shows that as time has progressed, this loudness level has increased. This is important for artists to understand in order to stop this phenomenon. It may also lead to better performance as lowering this loudness can lead to better sounding tracks that contains more detail.

If steps are followed to recreate some of the elements in popular songs or build off of them, the possible popularity of this track can be projected. The genre and similar songs can also be identified and organized automatically. This leads to the betterment of artists making music, as well as users looking for recommendations for new music on Spotify.

## ACKNOWLEDGMENT

I would like to acknowledge the help of Weijie Pang in the creation of this paper. Their contributions to research methods and overall organization of the report are greatly appreciated.

## REFERENCES

[1] J. Arvidsson, "30000 Spotify Songs," Kaggle, 2024. Available: https://www.kaggle.com/datasets/joebeachcapital/30000-spotify-songs

[2] Spotify, "Web API • References / Tracks / Get Track," 2025. Available: https://developer.spotify.com/documentation/web-api/reference/get-track. [Accessed: 2025]