**Spotify understand which features make a song popular?**

Heather Lemon

Department of Computer Science, University of Denver

COMP 4449 Capstone Midterm

Dr. Claudio Delrieux

April 1, 2023

**Spotify understand which features make a song popular by Heather Lemon**

Spotify is a popular music and media steaming service. Founded in 2006, it has grown to over 450 million active users and is one of the most successful and largest music streaming platforms available today.[2]

**Description of the Dataset**

The dataset comes from five files. The ranking\_spotify.csv files contains more than 2 million rows, which comprises 6629 artists, 18598 songs for a total count of one hundred five billion streams count. This csv contains the daily ranking of the 200 most listened songs every day, in 53 countries from 2017 and 2018 in Spotify.

For the purpose of narrowing the scope of the dataset, we will choose only songs from the year 2017 and country.

The second dataset comes from featuresdf.csv. It contains popular song attributes such as (tempo, energy, song name etc.) There are 13 numerical columns about a song's attributes and 3 categorical columns (song name, artist, key symbol (C#))[1]. By scoping the issue down to only songs from region USA and year 2017, we have 72,400 rows from the daily popular song list.

**Data Preprocessing**

The first step was cleaning and feature engineering. Where I chose to covert the time in milliseconds to minutes/seconds, a much more human readable format. The other feature was converting the key signature from 0-11 to the symbol used such as C#.

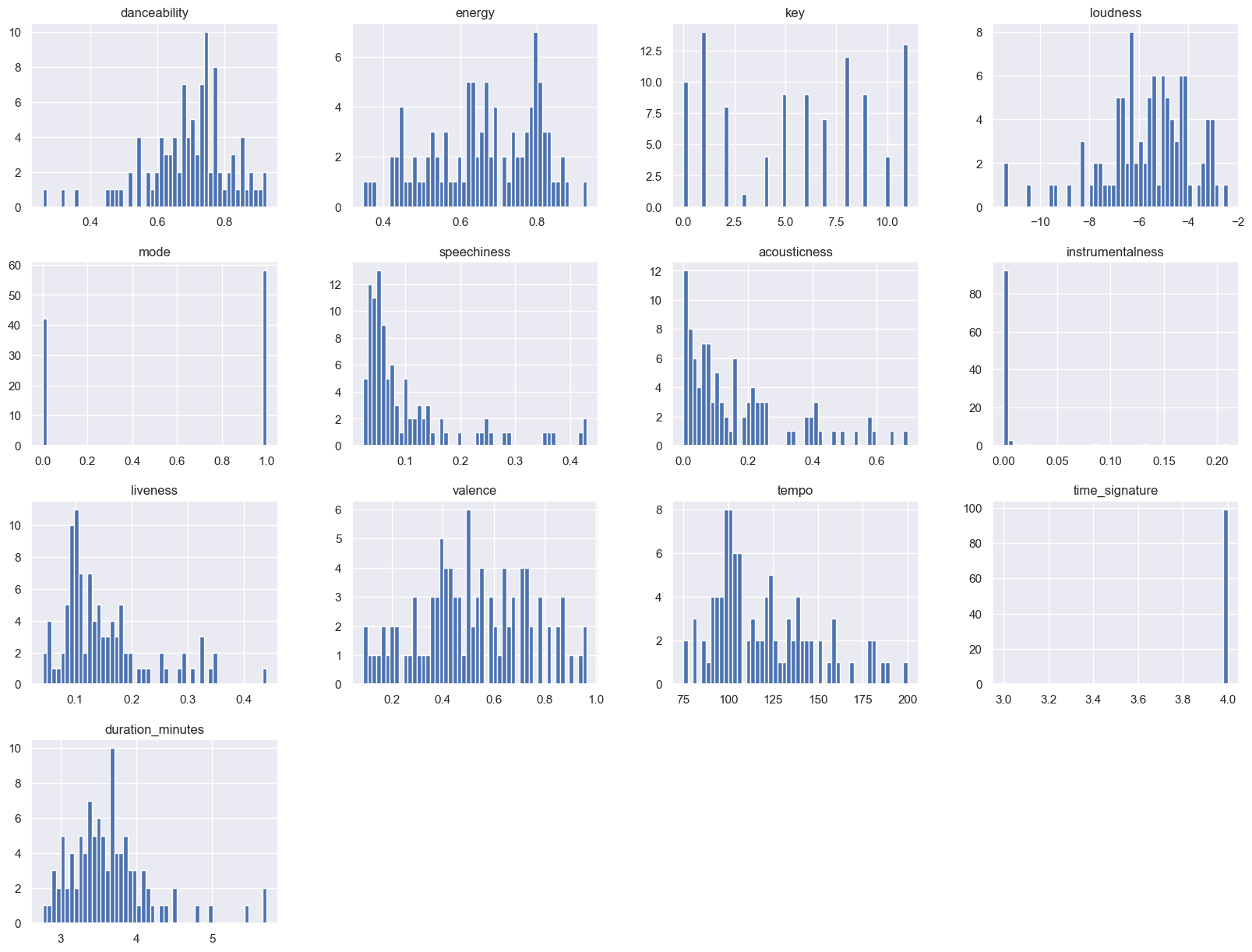
Null checks and data type checks were implemented on both dataframes; ranking\_spotify\_1 and features\_df.

**Exploratory Data Analysis**

Seventeen features is a lot to show for a pairplot, so I went with a histplot of the features df.

**Figure 1**

*HistPlot of all numerical features df*



*\*Note:* *There is a mix of normal and non-normal distributions, some have a normal distribution while other features only have one to two values.*

**Figure 2**

*Statistical Output of Song Details*

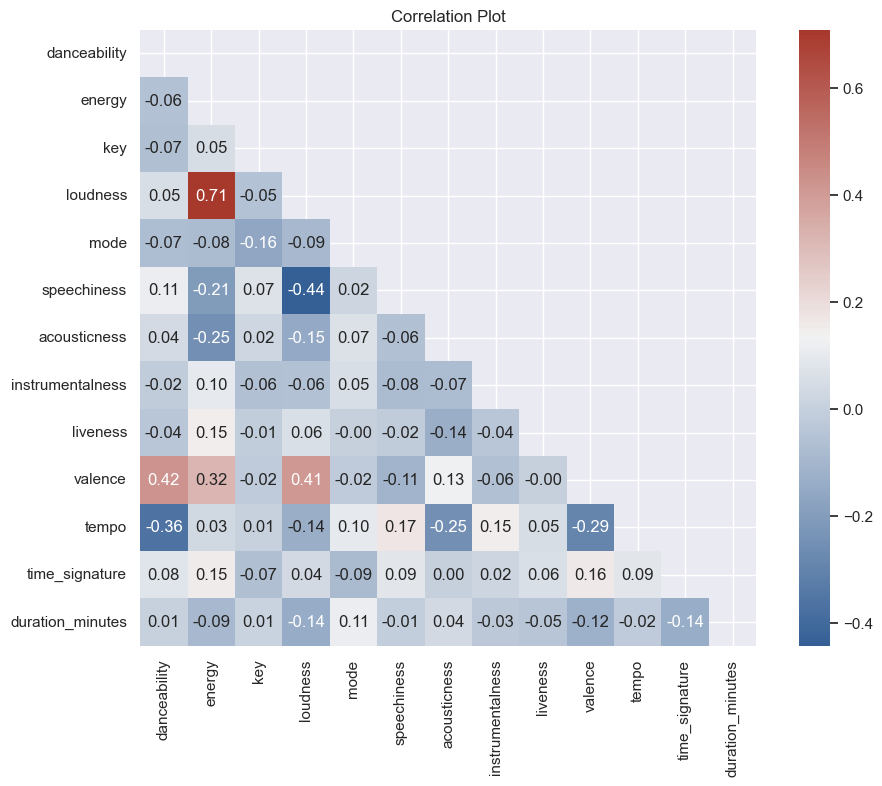
*Graphical user interface, text

Description automatically generated*

*\*Note: Most features between mean and median are similar but a few of the features are skewed and the standard deviation is low except for tempo (BPM).*

**Figure 3**

*Correlation Plot*

****

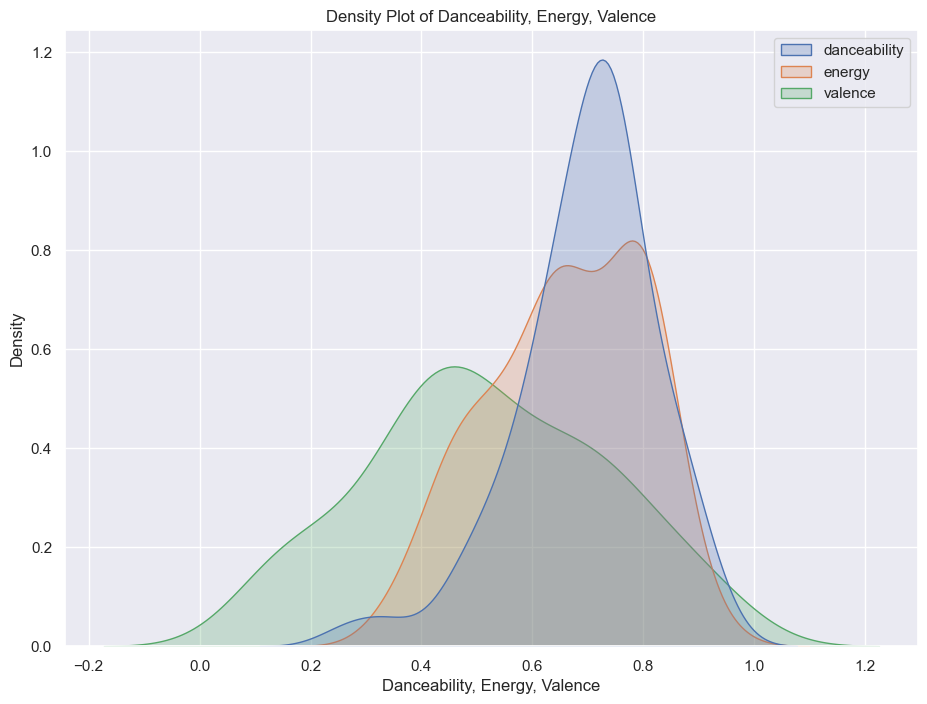
*\*Note: Energy and loudness have a strong positive correlation.*

*Valance and danceability, energy and loudness all have a positive correlation.*

*Loudness and speechiness have a negative correlation, which makes sense if a song is very loud you likely cannot hear the person singing in the background unless you're talking about death metal :) Surprising note - I expected danceability to have a stronger correlation with loudness, but there can be songs to be used to dance formally like a waltz or tango.*

**Figure 4**

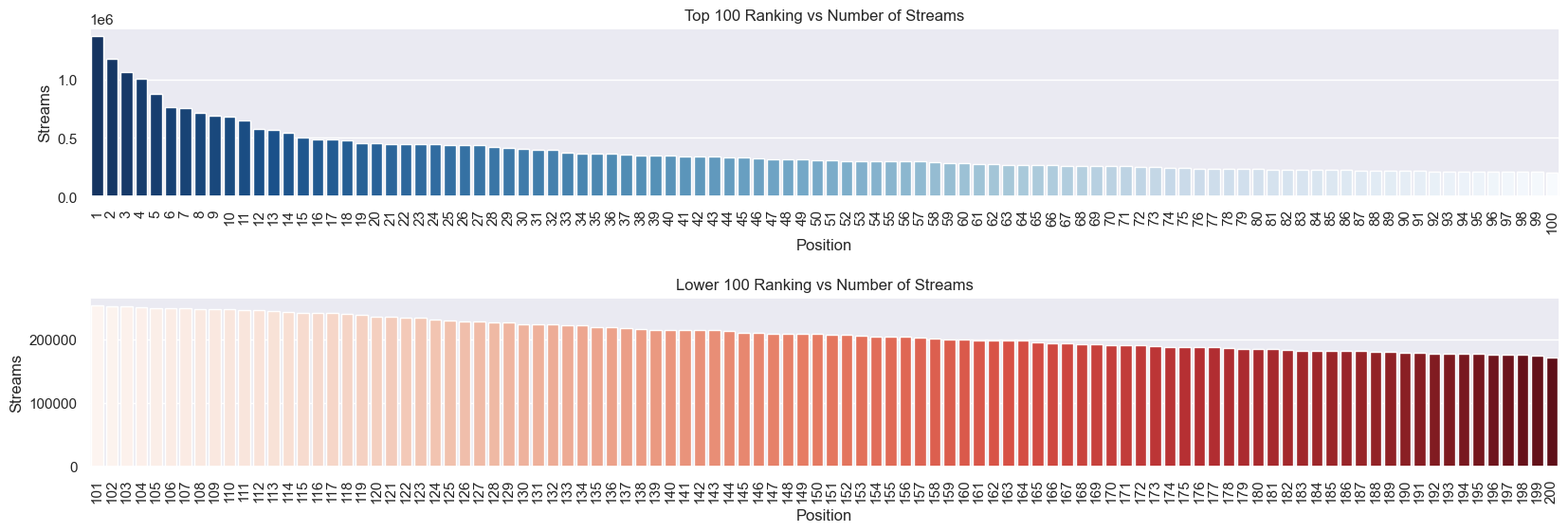
*KDE Plot of positively correlated features*



*\*Note: the shapes are similar*

**Figure 5**

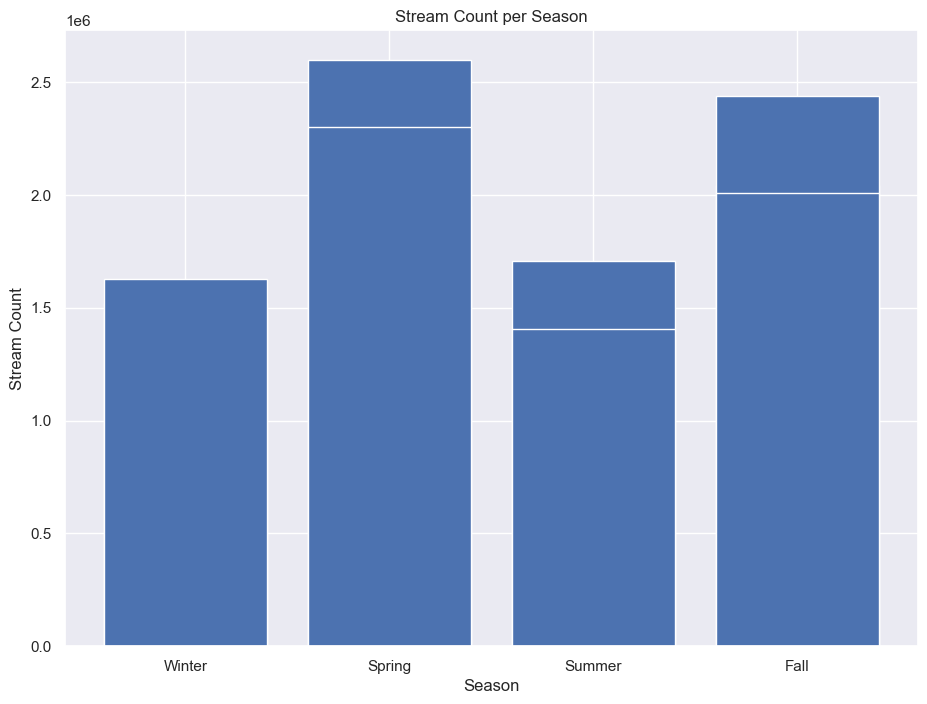
*Top 100 Rankings vs Number of Streams*

**

*\*Note: The top 100 most stream songs with the highest 5 having over a million streams per song with the lower 25% still around 500k streams. While the lower 100 rankings have values around 200k.*

**Figure 6**

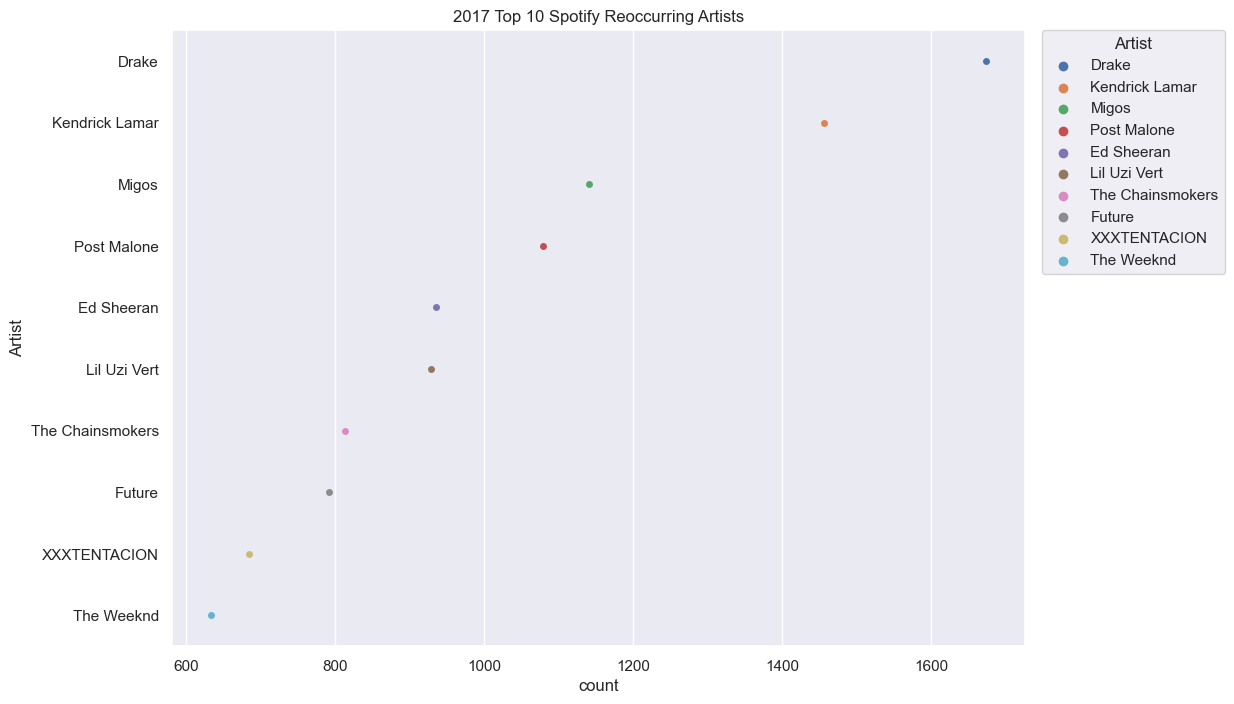
*Stream Count per Season*

**

*\*Note: Spring had the highest average streamer count*

**Figure 7**

*Top 10 Spotify Reoccurring Artists*



*\*Note: This was just for fun, but it seems Drake is pretty popular*

A key feature to this project is using features\_df as a lookup table for popular song details.

We merge the two dataframes together to include popular song details via song and artist name.

We drop the categorical features before the modeling phase including anything that was not part of the original dataset, i.e. We don’t want to include the month in song attributes list.

Data scaling; we did infact use the StandardScaler method because the features all have different relational values i.e. energy is not on the same scale as minutes.

Data Split; I used an 80/20 split of the data leaving us with (16184, 13) training examples and (4047,) test examples.

**Model Building and Evaluation**

\* Linear Regression

\* Decision Tree Regressor

\* Random Forest Regressor

\* SVM-R

\* Bagging Regressor

\* Lasso Regressor

\* Ridge Regressor

**Figure 8** *Linear Regression Coefficients*

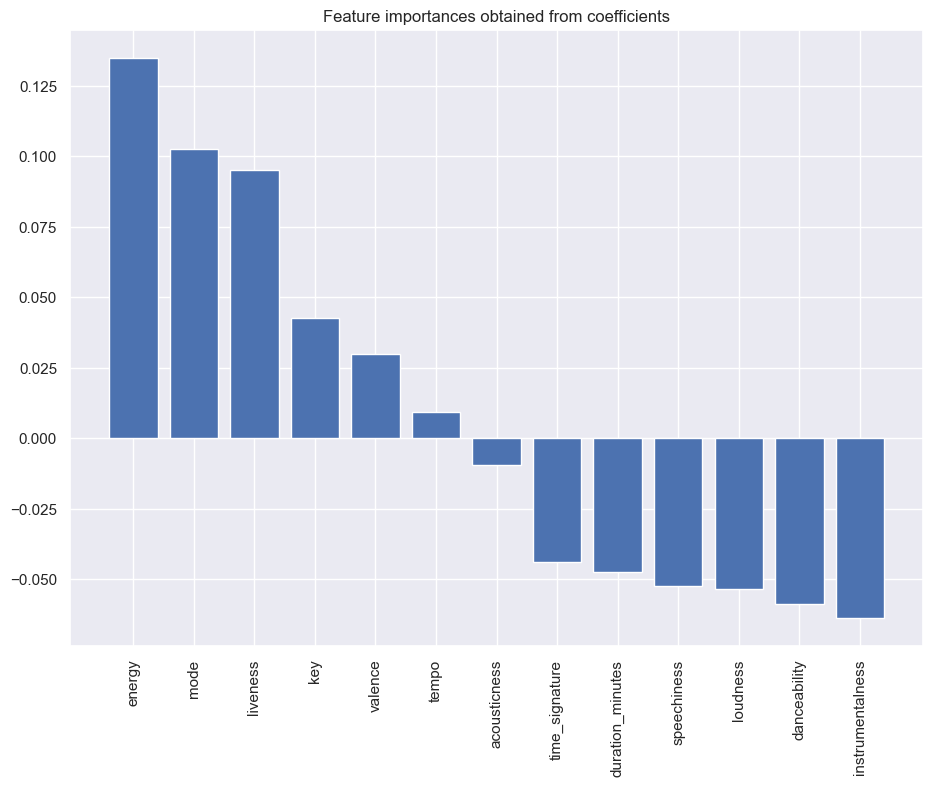
Graphical user interface

Description automatically generated

*Energy was the most important feature from our Linear Regression model*

**Figure 9**

*Linear Regression Feature Importance*

**

**Figure 10**

*OLS Summary*

Table

Description automatically generated with medium confidence

The R^2 value is 0.05% which explains nothing about the data. There are several factors that have a p-value less than 0.05 indicating that the features do have a significant impact on popularity.

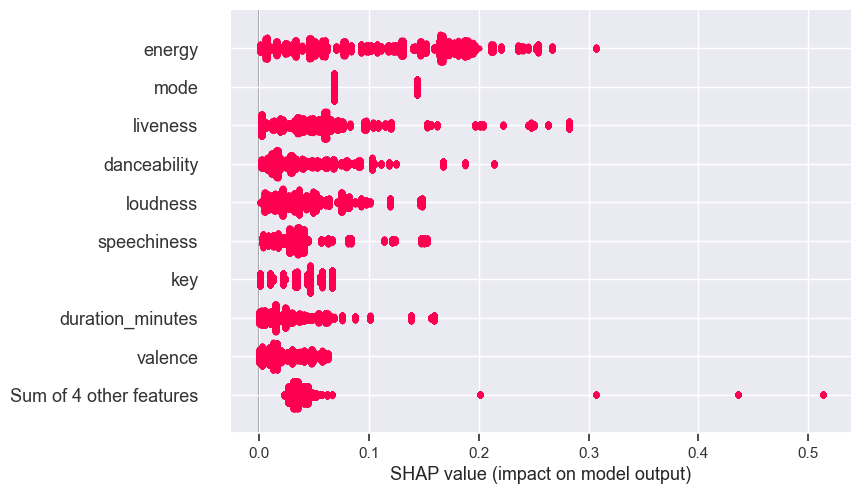
The linear regression model and OLS states that energy was the most important factor in determining popularity of a song. \*\* Remember this as we will see it again soon.

**SHAP and Explainable AI**

For machine learning models this means that SHAP values of all the input features will always sum up to the difference between baseline (expected) model output and the current model output for the prediction being explained.

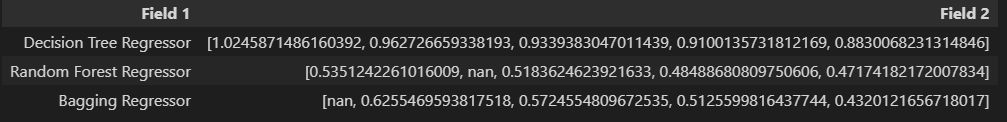
**Figure 11**

*Beeswarm plot*



*\*Note: Beeswarm plot shows the same explainable output for the linear regression model output of feature importance*

Figure 5 *Cross Validation Scores*



The Decision Tree Regressor is sill the best performing model out of the top three.

After running the GridSearchCV and tuning the hyperparameters that looks like this

**Figure 6** *Hyperparameter Tuning*

Text

Description automatically generated

*Hyperparameter Tuning Decision Tree Regressor*

**Final Model Selection**

Model selection

**Conclusion**

#TODO In answering the question which features best predict what makes a song popular.

**Surprise! Explainable machine learning with explainerdashbord package**

Text

Description automatically generated

The explainerdashboard package only need the machine learning model, the training or test set and to have an Explainer variable instantiated which we already did for the linear regression model and so we will use the lr\_model as an example here. Clicking the link will open a new browser tab to show you the dashboard the model built.

**Lessons Learned**

<lesson learned>

**Future Improvements**

Although we were unable to use any of the textual data,

**References**

\* [0] https://developer.spotify.com/documentation/web-api/reference/#/operations/get-several-audio-features

\* [1] https://en.wikipedia.org/wiki/Pitch\_class

\* [2] https://en.wikipedia.org/wiki/Spotify

\* [3] https://www.kaggle.com/datasets/edumucelli/spotifys-worldwide-daily-song-ranking

**Glossary Terms**

|  |  |
| --- | --- |
| Term | Description |
| id | id of song |
| name | Name of song |
| Artists | Song artist name |
| Danceability | 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. |
| 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. |
| key signature | The key the track is in. Integers map to pitches using standard Pitch Class notation.[1] E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on. If no key was detected, the value is -1. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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). |
| 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. |
| duration\_ms | song length in milliseconds |
| 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". |

\* Mood: Danceability, Valence, Energy, Tempo

\* Properties: Loudness, Speechiness, Instrumentalness

\* Context: Liveness, Acousticness