Executive Summary

Our data analysis report aims to analyze a public dataset from Kaggle called "Spotify and YouTube" to build supervised models that can predict the popularity of a song on Spotify. The primary objective is to determine the factors that contribute the most to the popularity of songs to help artists create well-received music.

The dataset consists of over 20,000 records, collected on February 07, 2023, and contains 27 variables such as track, artist, album, danceability, energy, key, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo, duration, stream, title, channel, views, likes, comments, description, licensed, official_video, and others.

The dataset from Kaggle will be preprocessed and analyzed using tools such as Python, and its libraries, such as Pandas, NumPy, Matplotlib, and Scikit-Learn. Jupyter Notebook is the primary tool for coding, documentation, and visualization.

We will use four supervised regression models, including Linear Regression, Ridge Regression, Decision Tree, and Random Forest, to predict the popularity of songs on Spotify, using the number of streams as the dependent variable. This will be accomplished by using statistical and machine learning techniques and identifying the most important variables that affect song popularity.

The Random Forest Model performed the best, with an R^2 value of 0.52. The factors that revealed the most impact on the number of Spotify streams include Likes, Views, Single, Acousticness, and Licensed. The most popular songs seem to be those with a larger number of likes, not released as a single, and do not have an acoustic sound.

Variables Description

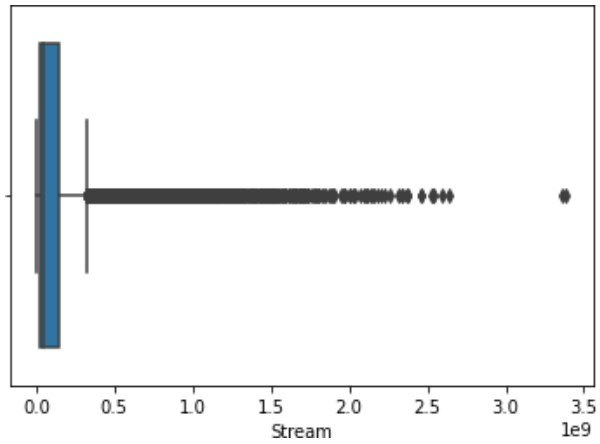
Name	Definition
Stream	number of streams of the song on Spotify (dependent variable)
Track	name of the song, as visible on the Spotify platform
Artist	name of the artist
Url_spotify	the URL of the song
Album	the album in which the song is contained on Spotify
Album_type	indicates if the song is released on Spotify as a single or contained in an album
Uri	a Spotify link used to find the song through the API
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.

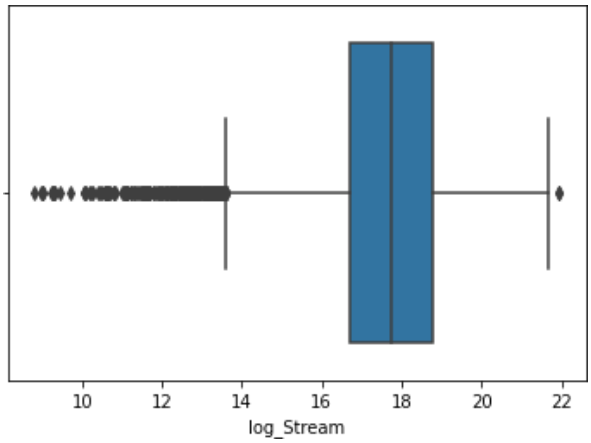
Energy	a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
Key	the key the track is in. Integers map to pitches using standard Pitch Class notation. 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 correlation of physical strength (amplitude). Values typically range between -60 and 0 db.
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 a 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	the duration of the track in milliseconds

Url_youtube	URL of the video linked to the song on YouTube, if any
Title	title of the videoclip on YouTube
Channel	name of the channel that has published the video
Views	number of views of video on YouTube
Likes	number of likes of video on YouTube
Comments	number of comments of video on YouTube
Description	description of the video on YouTube
Licensed	Indicates whether the video represents licensed content, which means that the content was uploaded to a channel linked to a YouTube content partner and then claimed by that partner.
Official_video	Boolean value that indicates if the video found is the official video of the song

Data Preparation

The data cleaning process consisted of using Python to remove unnecessary columns and null values within the data, which reduced the dataset to 19,170 rows and 14 columns. To evaluate the skew, if any, for our dependent variable, Stream, we created a boxplot. This boxplot showed us that the data is significantly right skewed. To counteract this issue, we took the log transformation of the y variable and produced another boxplot, which corrected the data skew, but left many outliers. Since outliers consisted of less than 2% of the log transformed data, we decided to remove them, reducing the number of rows to 18,871.



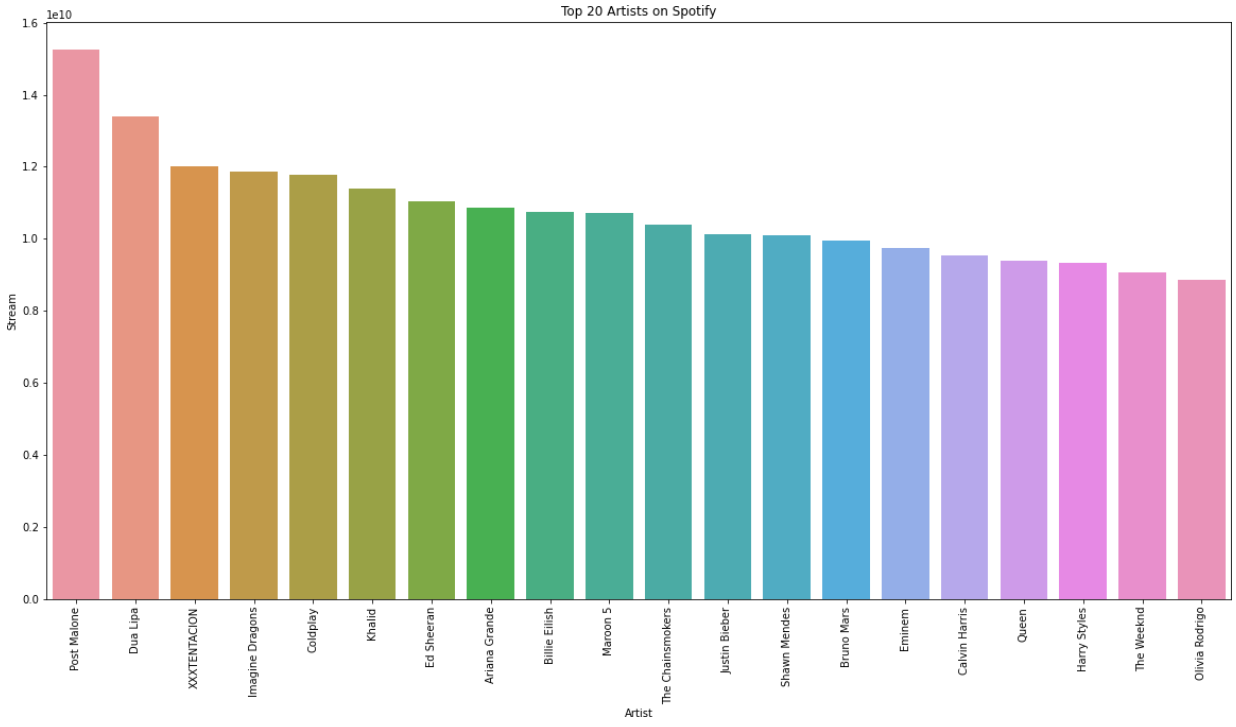


The VIF was calculated to determine which variables are highly correlated, if any. The output showed that the variables, Danceability, Tempo, and Energy are highly correlated with a VIF greater than 10; therefore, we removed Danceability and Energy, which removed potential multicollinearity bias.

	feature	VIF
0	Loudness	7.336951
1	Danceability	16.578484
2	Key	3.152742
3	Speechiness	2.133395
4	Instrumentalness	1.691045
5	Liveness	2.474909
6	Valence	8.342606
7	Tempo	14.964459
8	Duration_ms	3.950475
9	Likes	8.833829
10	Views	6.368549
11	Comments	1.987581
12	Acousticness	3.187460
13	Energy	17.744014

	feature	VIF
0	Loudness	6.468906
1	Key	3.033970
2	Speechiness	1.984689
3	Instrumentalness	1.614067
4	Liveness	2.261603
5	Valence	5.316499
6	Tempo	9.832408
7	Duration_ms	3.666180
8	Likes	8.748916
9	Views	6.356726
10	Comments	1.980150
11	Acousticness	2.921054

To determine the top contenders of popular songs on Spotify and their stream quantity, we ran a vertical bar graph with the top twenty artists in descending order, as shown below.



Models & Optimization

We focused our efforts on creating supervised regression models to predict the number of streams of a given Spotify song. Four models were created and evaluated, including Linear Regression, Ridge Regression, Decision Tree, and Random Forest. Before starting with the first model, linear regression, we created dummies for the categorical variable, Album_type and assigned independent and dependent variables to X and y, respectively. Since there is potential for skewness issues, we made sure to scale the data to normal distribution using StanderdScalar. The data was then separated into training and testing sets with a 70/30 split.

Linear Regression Model

The linear regression model is as follows:

log(Stream) = 0.128 * Loudness – 0.004 * Key – 0.050 * Speechiness – 0.096 * Instrumentalness – 0.071 * Liveness – 0.075 * Valence + 0.016 * Tempo – 0.019 * Duration_ms + 0.962 * Likes + 0.033 * Licensed – 0.130 * Views – 0.418 * Comments – 0.080 * Acousticness – 0.293 * Single – 0.037 * Compilation + 17.737

	coef	std err	t	P>|t|	[0.025	0.975]
const	17.7365	0.011	1580.697	0.000	17.715	17.758
Loudness	0.1276	0.016	8.022	0.000	0.096	0.159
Key	-0.0036	0.011	-0.325	0.745	-0.026	0.018
Speechiness	-0.0497	0.011	-4.359	0.000	-0.072	-0.027
Instrumentalness	-0.0960	0.014	-7.064	0.000	-0.123	-0.069
Liveness	-0.0710	0.011	-6.384	0.000	-0.093	-0.049
Valence	-0.0746	0.012	-6.214	0.000	-0.098	-0.051
Tempo	0.0159	0.011	1.388	0.165	-0.007	0.038
Duration_ms	-0.0194	0.010	-1.880	0.060	-0.040	0.001
Likes	0.9615	0.033	29.101	0.000	0.897	1.026
Licensed	0.0327	0.011	2.856	0.004	0.010	0.055
Views	-0.1302	0.027	-4.808	0.000	-0.183	-0.077
Comments	-0.4177	0.024	-17.766	0.000	-0.464	-0.372
Acousticness	-0.0799	0.014	-5.908	0.000	-0.106	-0.053
Single	-0.2931	0.012	-25.434	0.000	-0.316	-0.271
Compilation	-0.0366	0.011	-3.232	0.001	-0.059	-0.014

Looking at the p values from the OLS Regression, the variables Key, Tempo, and Duration_ms are not statistically significant for alpha = 0.05. However, since Duration_ms is very close with a p-value of 0.06, we decided to keep this variable. We dropped Key and Tempo and re-ran the Linear Regression model.

	coef	std err	t	P>|t|	[0.025	0.975]
const	17.7364	0.011	1580.779	0.000	17.714	17.758
Loudness	0.1289	0.016	8.116	0.000	0.098	0.160
Speechiness	-0.0489	0.011	-4.301	0.000	-0.071	-0.027
Instrumentalness	-0.0962	0.014	-7.085	0.000	-0.123	-0.070
Liveness	-0.0710	0.011	-6.379	0.000	-0.093	-0.049
Valence	-0.0742	0.012	-6.185	0.000	-0.098	-0.051
Duration_ms	-0.0197	0.010	-1.908	0.056	-0.040	0.001
Likes	0.9613	0.033	29.094	0.000	0.897	1.026
Licensed	0.0327	0.011	2.857	0.004	0.010	0.055
Views	-0.1305	0.027	-4.817	0.000	-0.184	-0.077
Comments	-0.4172	0.024	-17.744	0.000	-0.463	-0.371
Acousticness	-0.0809	0.014	-5.990	0.000	-0.107	-0.054
Single	-0.2933	0.012	-25.461	0.000	-0.316	-0.271
Compilation	-0.0366	0.011	-3.230	0.001	-0.059	-0.014

In the new model, all variables are statistically significant, (Duration_ms is again very close with a p-value of 0.056). The coefficient magnitudes have not changed much and the signs (positive/negative) for each respective variable are the same. Here we see that Loudness, Likes, and Licensed have a positive effect on the number of Streams a song receives, whereas Speechiness, Instrumentalness, Liveness, Valence, Duration_ms, Views, Comments, Acousticness, Single, and Compilation have a negative effect on the dependent variable.

Modified Linear Regression Model:

log(Stream) = 0.129 * Loudness – 0.049 * Speechiness – 0.096 * Instrumentalness – 0.071 * Liveness – 0.074 * Valence + 0.016 * Tempo – 0.020 * Duration_ms + 0.961 * Likes + 0.033 * Licensed – 0.131 * Views – 0.417 * Comments – 0.081 * Acousticness – 0.293 * Single – 0.037 * Compilation + 17.736

A few interesting points can be noted here:

Speechiness shows that songs will get more streams than audio books/talk shows/podcasts/etc. Valence shows us that more positive, cheerful songs have less streams than those with a low valence, sad songs.

Although the number of likes on YouTube has a positive correlation with the number of streams on Spotify, the number of views and comments on YouTube are actually negatively correlated with the dependent variable. This is likely solved by including non-linear terms for the Views and Comments variables.

The Single and Compilation variables show a negative coefficient, meaning songs that are released in an album format have a high number of streams when compared to singles and compilations.

However, upon evaluation, the R^2 value is close to 0.26 for both the training and testing data for both linear regressions, meaning in these models, the explanatory variables are only able to explain 26% of variation in the dependent variable.

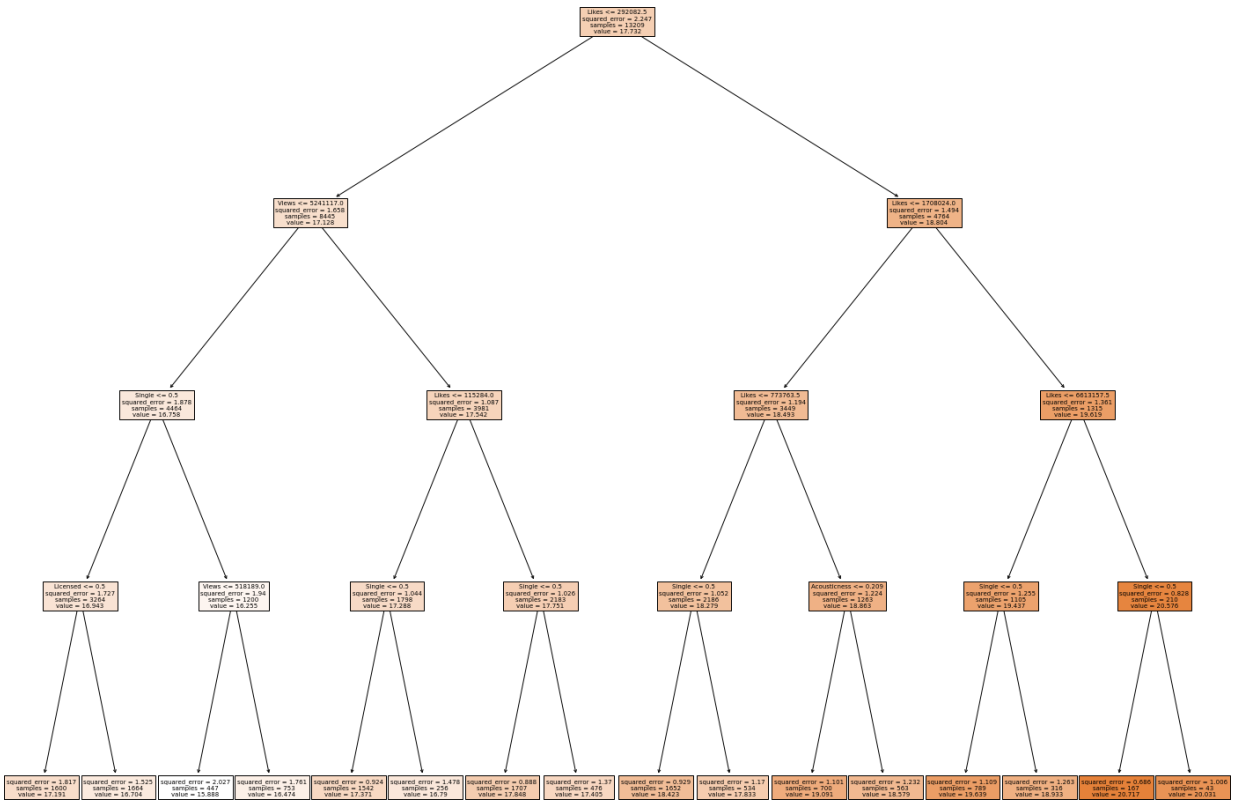
Ridge Regression Model

Our next approach was to use Ridge Regression since we have many x variables with small effects on the y variable. Since this model has a hyperparameter lambda which applies a penalty if there are unnecessary explanatory variables, we will be able to regularize the model and avoid overfitting issues. Using the grid search method, we chose alphas of 0.00001, 0.0001, 0.001, 0.01, 1, 10 and performed cross validation with 5 folds.

After evaluating for the best performance measure, alpha of 0.00001 was the best parameter with the score of 0.26. The R^2 value for the training and testing data was still around 0.26, like the linear regression model. Even with this model, the explanatory variables are only able to explain 26% of variation in the dependent variable.

Decision Tree Model

Since the linear regression model and ridge regression model with penalty was not able to explain much of the dependent variable variation, our next idea was to use a more advanced model, the Decision Tree. After a process of trial and error, the regressor model was fitted with a max depth of 4, which gave an R^2 value of 0.431.



After plotting this tree, we see that the variables with the most impact on number of streams include, Likes, Views, Single, Acousticness, and Licensed. The songs with the highest predicted number of streams seem to be those with higher amount of likes, less views on YouTube, licensed, not a single, and less acousticness.

Upon evaluation using the grid search method, the best max depth value is 6, with an R^2 value of 0.435. This is so far the best model, as it can explain 44% of the variation in the dependent variable.

Random Forest Model

Our final approach to creating an effective model was the Random Forest method. We chose this model because it is proven to have good results and accuracy for various problems and gives a more stable and better generalization when compared to the decision tree regressor. Although Random Forest will take longer to compile, due to its complexity, it is likely to have a lower bias and variance than the other models we created.

We set the max depth values between 2-14 and performed the model with grid search. The grid search method suggested max depth of 14 as the best model for the testing set with an R^2 value of 0.517; however, the training data for this depth would be much more overfitted with an accuracy of approximately 0.837. Therefore, we decided that the max depth of 12 was a better model, with training data accuracy of about 0.768, and testing data R^2 value of 0.522. We tried to close the gap between training and testing set accuracy by trying different numbers of cross-validation folds; however, 5 folds gave the best model. Although there is still an overfitting issue, this final model is able to explain 52% of the variation in the y variable, which is significantly higher than the other three models.

Limitations & Enhancements

Although the Random Forest Regressor brought accuracy up to 52%, we are still unable to explain 48% of the variation in the number of streams on Spotify for this dataset. The low R^2 could be a result of a biased dataset. The “Spotify and YouTube” dataset may have compiled songs that are all along the same range of popularity, creating an overfitting issue. There is also a high chance of omitted variable bias. Given additional significant explanatory variables, we could potentially improve the accuracy of the model. For example, the genre of the song, whether the comments are disabled on the YouTube video, whether the comments were positive or negative, the release date of the song on Spotify, whether the song was released before or after Spotify existed, and the language of the song. Additionally, more advanced models, like Neural Network, can be implemented, to account for the variation of the dependent variable which the other models could not explain.

Conclusion

In conclusion, this data analysis report focused on analyzing the factors that contribute to the popularity of songs on Spotify using a public dataset from Kaggle called "Spotify and YouTube". The data was preprocessed and analyzed using Python and its libraries, and four supervised regression models were used to predict the number of streams of songs on Spotify. The Random Forest Model performed the best, and the factors that revealed the most impact on the number of Spotify streams include Likes, Views, Single, Acousticness, and Licensed. The limitations of the data include not having enough significant explanatory variables in the data and a potentially biased dataset. Overall, this analysis provides valuable insights for artists and music industry professionals to create well-received music on Spotify.