Spotify Tracks Popularity

ISYE 7406: Data Mining & Statistical Learning

Submission Date: 4/10/2024

Kelly Lam (Student ID XX502 , klam62@gatech.edu)

Felipe Gastaldi (fgastaldi3, fgastaldi3@gatech.edu)

Trang Doan (tddoan64@gatech.edu)

Betsy Lee (Student ID XX421, blee452@gatech.edu)

Abstract

This report explores predicting the popularity of a variety of Spotify songs using audio features as predictors. The goal is to statistically discover the song characteristics that contribute to a track's popularity, potentially aiding artists and producers in developing more mainstream music. The project employs several statistical and machine learning models, such as logistic regression, linear discriminant and quadratic discriminant analysis, Lasso, Ridge regression, Naive Bayes, random forest, XGBoost, and Generalized Boosting models which are evaluated by accuracy, sensitivity, specificity, and F1 score. Monte Carlo cross-validation is also utilized to assess whether a model can generalize unseen data. However, the analysis shows that the models have signs of overfitting resulting in unreliable inferences.

Introduction

Spotify stands as a prominent music streaming platform that empowers artists and producers worldwide to showcase their creations to broader audiences. It revolutionizes our daily interaction with music, and it's remarkable to observe the vast amount of data it generates. The goal of this project is to utilize several machine learning algorithms capable of predicting the potential popularity of new tracks based on certain attributes. The data comes from a Spotify Tracks database sourced from Kaggle which includes 114k unique tracks and accompanying attributes including tempo, loudness, duration, and several other song components. The intention is to develop an algorithm that best predicts popularity so that artists can understand the composition of best performing songs and get a sense of how a new song may perform. This could enable future artists or producers to create or adapt songs to be more appealing to the public.

We will use statistical methods such as checking for model assumptions, multicollinearity, and oversampling, along with models which include logistic regression, random forest, Generalized boosting, XGBoost, linear discriminant analysis, quadratic discriminant analysis, Naive Bayes, Lasso, and Ridge regression. We will use accuracy and precision and recall to evaluate the performances of each model on both the training/testing datasets as well as Monte Carlo cross-validation. The models will also be compared against each other to confirm the model with top performance.

Data Source

The dataset comes from Kaggle under the name "Spotify Tracks Dataset" and can be found at this link. There are 114,000 observations with 21 total columns (the following columns and descriptions can be found in the appendix).

Below, you can see the first three rows of our dataset prior to data cleaning.

X	track_id	artists	album_name	track_name	popularity	duration_ms	explicit	danceability	energy	key	loudness	mode	speechiness	acousticness
0	5SuOikwiRyPMVoIQDJUgSV	Gen Hoshino	Comedy	Comedy	73	230666	False	0.676	0.461	1	-6.746	0	0.1430	0.0322
1	4qPNDBW1i3p13qLCt0Ki3A	Ben Woodward	Ghost (Acoustic)	Ghost - Acoustic	55	149610	False	0.420	0.166	1	-17.235	1	0.0763	0.9240
2	1iJBSr7s7jYXzM8EGcbK5b	Ingrid Michaelson;ZAYN	To Begin Again	To Begin Again	57	210826	False	0.438	0.359	0	-9.734	1	0.0557	0.2100

instrumentalness	liveness	valence	tempo	time_signature	track_genre
1.01e-06	0.358	0.715	87.917	4	acoustic
5.56e-06	0.101	0.267	77.489	4	acoustic
0.00e+00	0.117	0.120	76.332	4	acoustic

The variable “Popularity” is the original ranking of the track in Spotify platform, ranging from 0 to 100, with 100 denoting an extremely viral song; for example, Unholy by Kim Petras featuring Sam Smith has a value of 100. According to Spotify, the popularity is calculated mostly based on the total number of plays and how recent those plays were. This means that popular songs that were played recently will have higher ranking than those that were played a lot in the past. We will use this variable as a filter to create our response variable. Observations with popularity ranking less than 80 will be labeled as non-popular or 0, while the rest will have the response equal to 1 to denote these observations as popular. Even though the cut-off seems significant enough to have a substantial number of observations in class 1, the truth is quite the opposite. There are only 1,201 observations having label as 1 based on this filter criteria out of 114,000 observations (~1.05%). This will be a challenging but very interesting factor in selecting and comparing model performance for this dataset.

We then drop irrelevant predictors such as ‘track id,’ ‘album name,’ and ‘track name’ because they are unique identifiers that are insignificant to the prediction power of our model and will cause expensive noise/time computation. ‘Artists’ and ‘track genre’ were then explored to identify if the values could be binned then converted to a factor variable. We find that there are 31,438 unique artist names and 114 unique musical genres with an equal distribution of 1,000 songs per genre. Due to ‘track genre’ being a categorical variable and having an equal distribution, we decide to drop this predictor from the dataset.

Due to ‘track genre’ being a categorical variable and having an equal distribution, we decide to drop this predictor from the dataset. Since we hypothesize that the attribute ‘artists’ has an impact on the popularity of a song, we considered applying the logic that if the artist of a song has a track popularity of 1, then the newly created predictor ‘popular artists’ would also be 1, else 0. However, this additional feature could potentially cause overfitting due to data leakage due to including the response into our independent variable. Therefore, we decided not to implement this feature in our models; we save the exploration of a combined dataset of gathering artists who have been in the Billboard Top 100 for the future. We decide to drop the column ‘artists,’ and we convert the Boolean predictor ‘explicit’ to numeric.

Exploratory Data Analysis (EDA)

The histogram for the popularity attribute (Figure 1) reveals that most observations are low in popularity, with about 87,000 observations having a popularity ranking of 50 and below. Only about 27,000 observations show songs with a popularity ranking of 50 or above. Due to the volume of the observations in the original dataset, we decided that subsampling is needed.

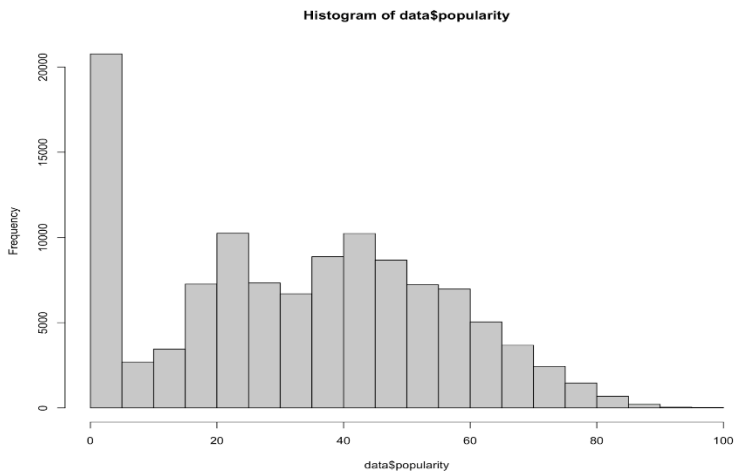


Figure 1 Histogram for " Popularity" Attribute

We continue our EDA by looking at the boxplots of several audio features grouped by non-popular and popular songs. The first one is **Duration.** The shorter the track is, the more popular it is.

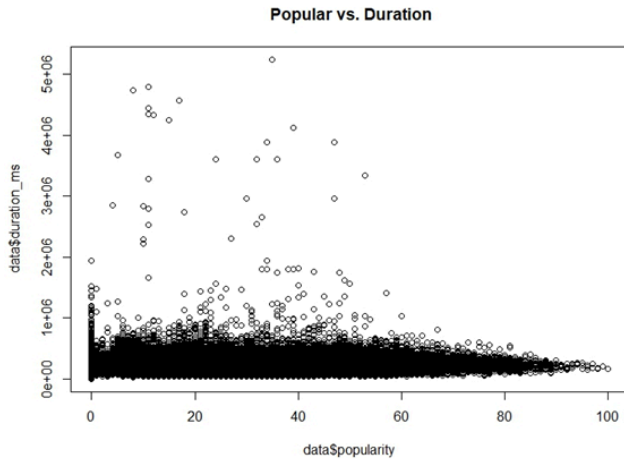


Figure 2: Popularity vs. Duration

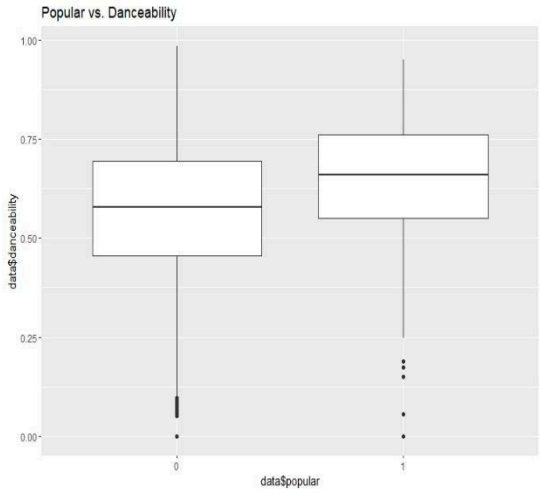


Figure 3: Popular vs. Danceability

Another variable that shows significant differences between the 2 groups is **Danceability**. This variable describes how suitable the track is for dancing and as you can see, the average danceability rating for popular songs tends to be higher than the non-popular songs.
A similar conclusion can be drawn for the **Valence** and **Energy** attribute as well. The popular group has higher valence on average. According to the attribute descriptions, higher valence songs are often happy and cheerful, bringing positive sound and experiences. These traits are often seen in high energy tracks as well.

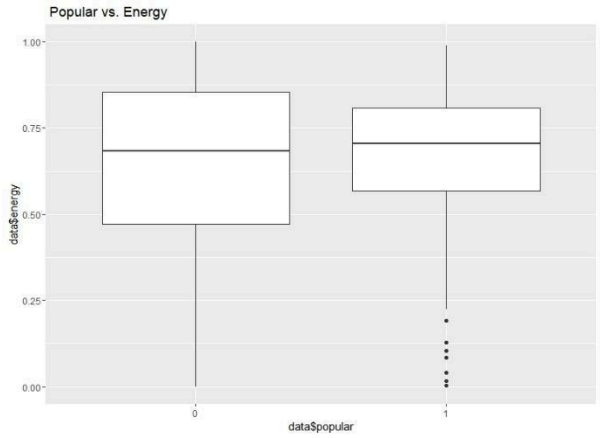


Figure 4: Popular vs. Energy

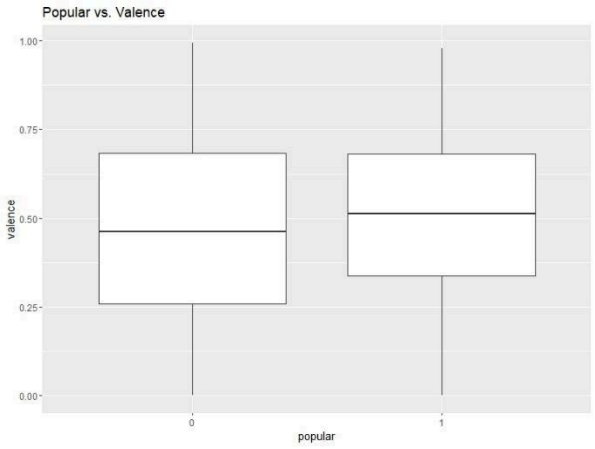


Figure 5: Popular vs. Valence

Since we observe the likelihood of popular songs having higher valence, energy and danceability, we would reasonably assume the same relationship would be for **Liveness** attribute. In fact, the opposite scenario happens here. A lot of unpopular songs have higher liveness even though the mean of liveness between 2 groups doesn't show significant difference.

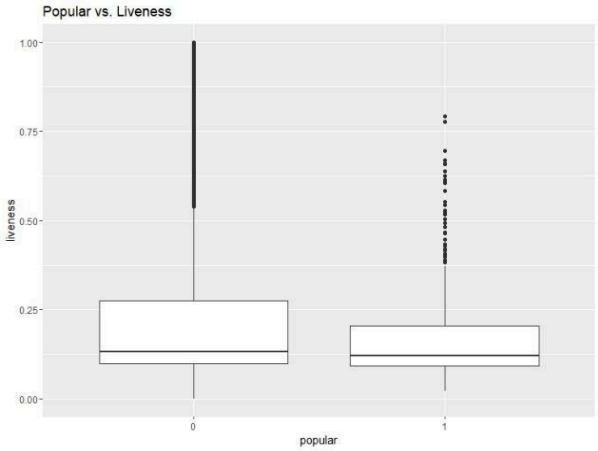


Figure 6: Popular vs. Liveness

Loudness may potentially affect song popularity since we observe here that majority of popular songs have slightly higher **loudness** ratings compared to non-popular songs.

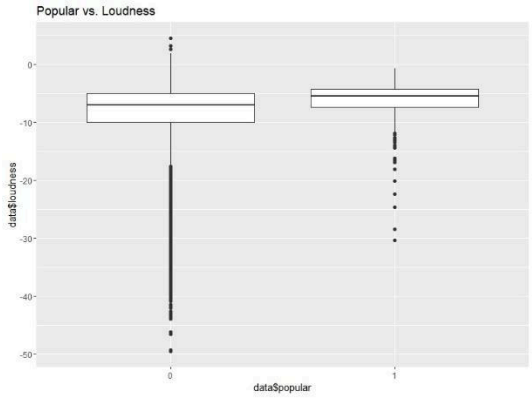


Figure 7: Popular vs. Loudness

The majority of popular songs have a lower **Acousticness** rating compared to non-popular songs. The median for **Acousticness** hovers around a rating of 0.169.

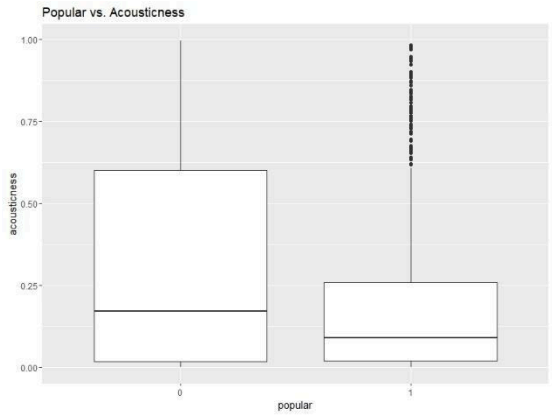


Figure 8: Popular vs. Accousticness

Similar to the **Acousticness** feature, the majority of popular songs have a lower **instrumentalness** rating compared to non-popular songs.

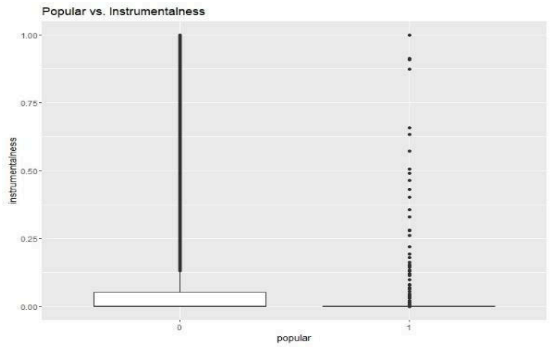


Figure 9: Popular vs. Instrumentalness

The bar plot in Figure 10 reveals that the overwhelming majority of songs do not contain **explicit** lyrics, indicating that ‘explicit' is not a relevant factor in predicting the popularity of a song.

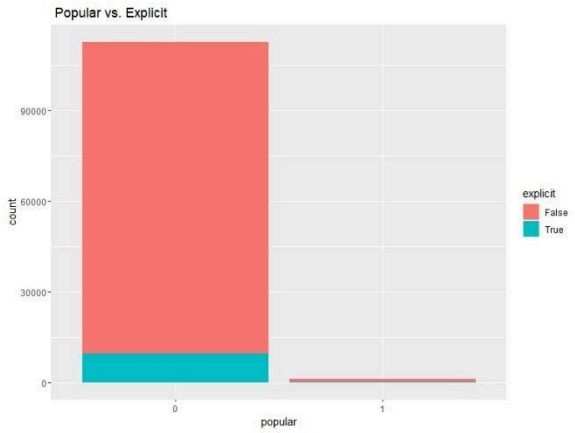
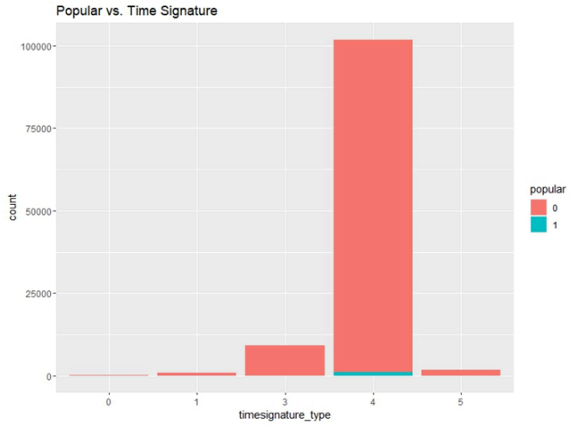


Figure 10: Popular vs. Explicit

Figure 11: Number of song per time signature types

Another interesting fact is that even though we have 5 different types of time signature, most of our observations have type 4. (101843 observations ~ 89.3%). Therefore, **timesignature_type** might not be a relevant variable here.



Additional graphic plots of other predicting variables can be found in Appendix 1.

To better understand the relationship between our response and independent variables, we plot the correlation matrix/heatmap (Figure 12) and focus on the bottom row . We find that there are extremely weak relationships between the predictors and the response. This could indicate that there is not enough information or that perhaps the data is not suited to predict the response variable, leading to poor modeling performance.

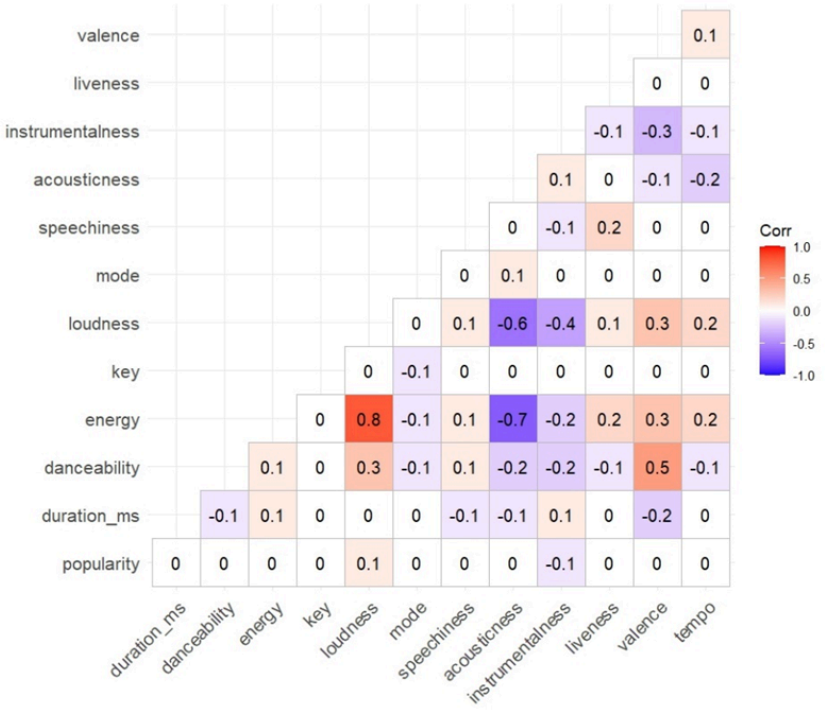


Figure 12: Correlation Matrix

The graph shows moderate positive correlation between valence and danceability, a strong positive correlation between loudness and energy, a strong negative correlation between energy and acousticness, as well as loudness, and acousticness, and weak or very weak correlation in the other predictors. To confirm there is no multicollinearity, which is suspected there will not be any, we use GVIF to calculate the corresponding values. We find that the GVIF values range from 1-2, indicating that there is no multicollinearity.

We also check the distribution of the response variable to understand how imbalanced the classes are. The majority class is the non-popular songs with 112,799 observations and popular songs with 1,201 observations. To mitigate this issue, we will explore different sampling techniques in the methodology section.

Proposed Methodology

Due to the combination of both numerical and categorical variables, we will perform the prediction using 9 different techniques:

1. **Logistic Regression**: This is considered the baseline model for classification problems. Since this is binary classification, we will need to set the cut-off rate manually or use the default rate of 0.5.
2. **Lasso Regression:** Least Absolute Shrinkage and Selection Operator (Lasso) regression is a regularization technique, guided by a selected lambda parameter. It uses shrinkage to decrease certain model coefficients to 0 in order to minimize the

impact on predictions, encouraging simple, sparse models. Can serve as a variable selection technique. Otherwise known as L1-norm regularization.

3. **Ridge Regression:** Similar to Lasso Regression, Ridge is a regularization technique that introduces a penalty term that guides the selection of coefficients for each predictor variable. Unlike Lasso, Ridge shrinks model coefficients but does not bring them to 0, it minimizes the reliance on individual predictors, thus it cannot serve as a variable selection technique. Otherwise known as L2-norm regularization.
4. **Naive Bayes**: Naive Bayes is a simple, probabilistic classification model based on applying Bayes' theorem for datasets with strong independence assumptions between the predictors, which we had established earlier that was the case for our features when we tested for multicollinearity.
5. **Linear Discriminant Analysis (LDA):** This technique originated from the Bayesian theorem to derive the discriminant functions and make classifications based on posterior probabilities. Here, we assume each response class would have normal distribution and identical covariance matrices. LDA constructs the decision boundaries with the goal of maximizing the between-class scatter and minimizing the within-class scatter.
6. **Quadratic Discriminant Analysis (QDA):** This is the special case of LDA in which the decision boundaries are quadratic instead of linearly produced by LDA and no assumption of identical covariance, thereby making QDA more flexible and better at modeling complex relationships.
7. **Random Forest**: Random Forest creates multiple subsets of the original dataset through bootstrap sampling. At each node of the decision tree, instead of considering all features to determine the best split, Random Forest selects a random subset of features, thus multiple decision trees are built independently, and the final prediction will be the combined predictions of all individual trees through a majority vote.
8. **XGboost**: Utilizing ensemble learning with re-weight technique, weak learners are combined to create stronger learners, XGboost builds trees sequentially where each tree corrects the errors made by the previous one.
9. **Generalized Boosting:** Similar to XGboost, Generalized boosting builds an ensemble of weak learners sequentially.

Before modeling, we split our dataset into training and testing sets. To reduce the effects of imbalances, we use stratified splitting, using 80% of the full data for training to keep the proportions of classes in the training and testing the same as the original dataset. In order to address the imbalanced class response issue for training, we use ROSE to oversample our data. Prior to oversampling, we have 90,244 observations in the majority class and 956 observations in the minority class. Post transformation, we get a total of 91,200 observations in the training set with 45,856 coming from the majority class and 45,344 from the minority class. To properly evaluate our model, we use accuracy along with sensitivity, specificity, and f1-score to determine which model best finds balance in class prediction.

To identify if logistic regression would be an appropriate choice for our dataset, we investigate model assumptions. We see that the qq-plot does not follow a normal distribution and the histogram is extremely skewed. Any statistical inference made using logistic regression would be unreliable because of the departures from model assumptions. Thus, we are unable to make a confident deduction about the model's significance using the chi-square test.

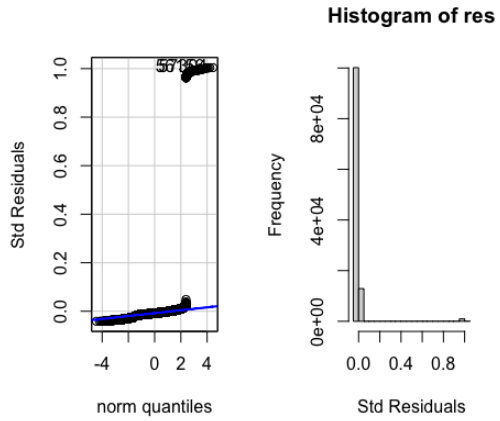


Figure 13: QQ-Plot and Histogram of Residuals

Despite departures from the logistic regression model assumptions, we want to evaluate the performance as a benchmark. We run logistic regression along with stepwise backward and forward regression for feature selection. Both stepwise models have the same AIC and features chosen. Here, a key difference is that the full logistic regression model has an AIC of 1619.8, and the reduced model which does not include features ‘valence’ and ‘time signature’ has an AIC of 1619.2 which is only 0.6 points less. A likelihood ratio test was then run for the two models with the null hypothesis that model 1 is sufficient in explaining the variability of the data compared to model 2. Here, we have model 1 as the reduced model and has p-value 0.2414 for the lrtest which suggests that the reduced model is sufficient. Multicollinearity and overdispersion was checked and found that none was present. The newly balanced training dataset was then used for the stepwise logistic regression.

In addition to Logistic Regression, we evaluated the performance of two regularization techniques, Lasso and Ridge regression. As can be seen above in the correlation matrix, there are no variables that are significantly correlated to popularity. Additionally, there exist very few impactful relationships between variables, with the only notable exceptions being energy-loudness, energy-acousticness, and loudness-acousticness. As a result, the optimal Lasso model selects all variables except for Key, directly resulting from the lack of significant influence of any particular predictors on the response variable. Similarly, the optimal Ridge model results in a significantly smaller coefficient for Key than any other predictor. Overall, we see fairly complete models that perform very well when predicting results for test data but have alarmingly low sensitivity scores, raising concerns around performance and widespread application.

The next two models are Linear Discriminant Analysis and Quadratic Discriminant Analysis. Two main assumptions for LDA and QDA are normal distribution and covariance. While LDA strictly requires normal distribution for each response class and identical covariance matrices to generate the discriminant linear function, QDA relaxes the equal covariance matrix assumption. We notice the variance, the length of each plot clearly differs in the exploratory data analysis, this is an indication of non-equal variances. Plotting the distribution for each variable, we can see how the normal distribution doesn’t hold as well. Below is an example of the density plot for liveness and energy.

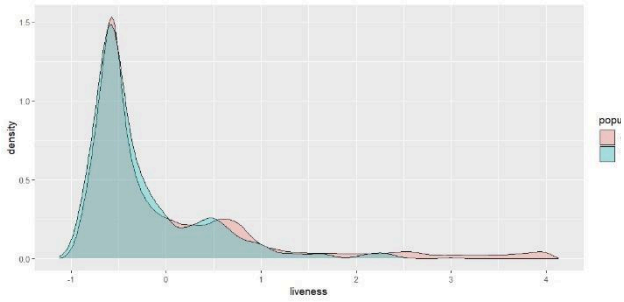


Figure 14: Liveness Density Plot

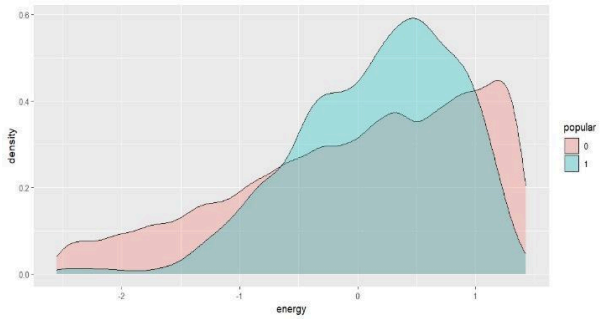


Figure 15: Energy Density Plot

Although we are publishing the result for the LDA & QDA methods, the dataset violates the model assumptions, thus, it might not be a good fit. The No information rate, the accuracy that would be achieved by always predicting the most frequent class in the training data, for QDA and LDA are consistently in the 95%+ range.Essentially, it represents the baseline accuracy that a model would achieve if it made the simplest possible prediction. Both LDA and QDA models can't achieve this high accuracy . This is where the significant imbalance dataset factor makes it very challenging for any model to “outsmart”.

We also run random forest using the default parameters and random forest using randomized search cross-validation to find the optimal number of variables to randomly sample as candidates at each split to maximize accuracy. We find that mtry = 14 and 500 trees was able to maximize accuracy for random forest. Due to the small size of our dataset, we further evaluate our models using Monte Carlo cross-validation. We take B = 50 and split the data into training and testing dataset in each loop, also using stratified splitting.

For a generalized boosting model, cross-validation helped us find the optimal number of iteration as 5000 trees with shrinkage value =0.01. Noticing how well random forest and generalized boosting models capture the minority observation correctly (all 4 metrics stay very consistently close), we decide to build a more complex model XGboost. When using default parameters, XGboost model shows a moderate performance among all metrics. We fine tuned it using cross-validation with 5000 iterations, nfold=5 and minimum child weight=2 . The training model later shows significant signs of overfitting with accuracy = 1, thus we decide to go with the default parameters.

Results and Analysis

Below is the summary table of all models that we ran.

Method	**Dataset**	**Accuracy**	**Sensitivity**	**Specificity**	**F1-Score**
Forward Stepwise Logistic Regression	Training (with oversampling)	0.689	0.647	0.732	0.676
	Testing	0.653	0.651	0.755	0.788

	Training (with Monte Carlo CV)	0.688	0.647	0.729	0.676
	Testing (with Monte Carlo CV)	0.652	0.651	0.739	0.787
Linear Discriminant Analysis	Training (with oversampling)	0.684	0.651	0.717	0.674
	Testing	0.688	0.657	0.720	0.681
	Training (with Monte Carlo)	0.683	0.650	0.717	0.674
	Testing (with Monte Carlo)	0.649	0.648	0.771	0.786
Quadratic Discriminant Analysis	Training (with oversampling)	0.704	0.505	0.906	0.632
	Testing	0.701	0.5	0.906	0.628
	Training (with Monte Carlo)	0.704	0.503	0.906	0.630
	Testing (with Monte Carlo)	0.421	0.415	0.932	0.587
Lasso	Training (with oversampling)	0.970	0.980	0.021	0.984
	Testing	0.951	0.962	0.000	0.975
	Training (with Monte Carlo CV)	0.907	0.916	0.006	0.951
	Testing (with Monte Carlo CV)	0.906	0.916	0.006	0.950
Ridge	Training (with oversampling)	0.966	0.976	0.025	0.982
	Testing	0.946	0.956	0.000	0.972
	Training (with Monte Carlo CV)	0.885	0.894	0.009	0.938
	Testing (with Monte Carlo CV)	0.885	0.894	0.008	0.938

Naive Bayes	Training (with oversampling)	0.964	0.145	0.973	0.253
	Testing	0.963	0.121	0.972	0.214
	Training (Monte Carlo CV)	0.958	0.154	0.967	0.265
	Testing (Monte Carlo CV)	0.955	0.196	0.963	0.326
Random Forest	Training (with oversampling)	1	1	1	1
	Training (with Monte Carlo)	0.858	0.813	0.905	0.858
	Testing	0.633	0.623	0.924	0.773
Generalized Boosting	Training (with oversampling and n.trees=5000)	0.781	0.738	0.825	0.772
	Testing	0.488	0.483	0.906	0.651
XGBoost (dropping: explicit, loudness,mode, time_signature)	Training (with oversampling)	0.680	0.944	0.413	0.748
	Testing	0.761	0.762	0.669	0.863

Our results show that XGBoost outperformed the other models. The results for XGBoost revealed a balanced performance across training and testing scenarios and generalizes well for unseen data. Due to the fact that the models exhibit signs of overfitting, excluding XGBoost, we forgo using the Wilcoxon test to compare model performances as there were poor performances across all the models.

Although Lasso, Ridge, and Naive Bayes demonstrated the highest rates of accuracy, either their sensitivity or specificity were unexpectedly low and raised some concerns on their model performances. The logistic regression model demonstrated a strong and balanced performance; however, model assumptions presented in the EDA section of our report, do not hold. Thus, we cannot ensure statistical significance of the results. The LDA and QDA models performed similarly to logistic regression. Although these models demonstrated a strong and balanced performance, their normality and variance assumptions do not hold. Random forest and generalized boosting yielded the highest performances across the four scores; however, these models also show significant signs of overfitting. Lastly, although XGBoost did not yield the highest accuracy rates, the model applied well to the testing dataset. This ensures the generalization capabilities of the model.

Conclusion

Based on the models run in this report, we find that there are clear signs of overfitting leading to unreliable inferences. These models are not able to capture the variability present in the data with the amount of information we supplied. Thus, we are unable to confidently conclude which predictors lead to popularity in songs. However, XGboost performs moderately well and would be the recommended model if further analysis is done.

Implication to the business:

1. The results from these tests can be applied by record labels or aspiring artists alike. While we have some hesitations with the results due to risks of over-fitting, the models developed can be utilized to create music that will reach a broad audience and by the parameters set out in this analysis, reach a considerable level of popularity. For example, when creating new music, artists can target specific values of each of the included predictors and get a sense of how the new music they are working on will perform once released to the general public on the Spotify platform. Assuming record labels and artists are focused on increasing reach and as a result, profits for new music, this pre-work can inform what type of songs will drive significant revenues and prevent time spent on songs that will not move the needle.

2. Imbalance dataset is a challenge for many models to learn from the minority class. It is important to recognize this trait early in the data exploratory stage and select splitting for training and testing accordingly to avoid extreme overfitting. This is crucial for models like LDA/QDA where the prior probability is calculated on the training set and the training set fails to represent the whole dataset.

3. Feature engineering is necessary. Even though we were unable to identify relevant variables, when using only selected variables in the XGboost models, we still observe low biased - decent performances across metrics with less variance.

4. In this class, we have studied the different ways of training, testing, and evaluating the performance of a variety of machine learning algorithms, both independently and compared to others. Additionally, we saw how depending on the algorithm selected for analysis and prediction, results may vary due to the differing optimization metrics utilized by each particular algorithm. The learnings from this class were applied directly to this project and as a result, we were able to represent the differing levels of predictive power and certainty based on the type of model selected as well as the characteristics of the dataset that is being analyzed.

Appendix

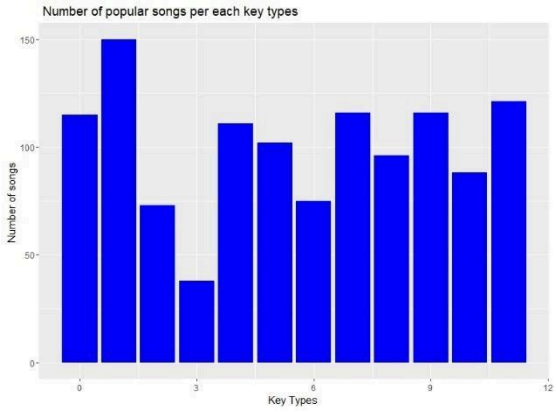
Attribute Description

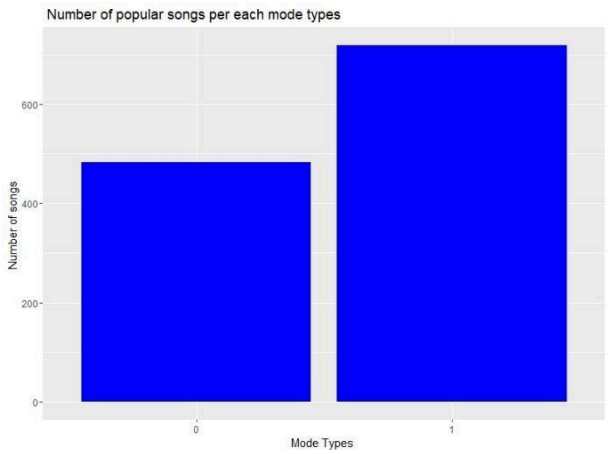
Attribute	Description	Value Range

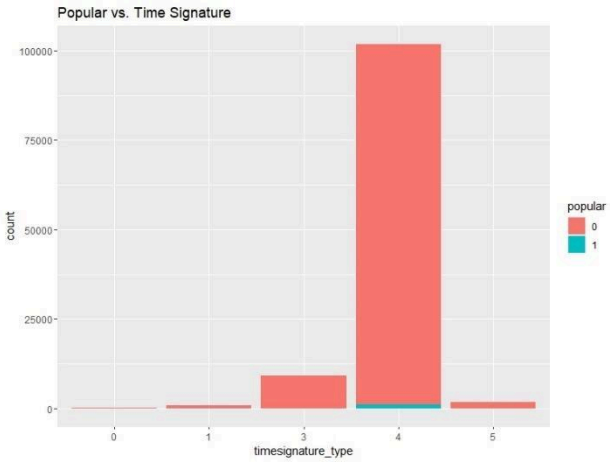
track id (character)	The Spotify ID for the track.	--
artists (character)	The artists' names who performed the track.	--
album name (character)	The album name in which the track appears.	--
track name (character)	Name of the track.	--
popularity (integer)	The popularity of a track is a value between 0 and 100, with 100 being the most popular. The popularity is calculated by algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are. Generally speaking, songs that are being played a lot now will have a higher popularity than songs that were played a lot in the past. Duplicate tracks (e.g. the same track from a single and an album) are rated independently. Artist and album popularity is derived mathematically from track popularity.	0 – 100
duration (integer)	The track length in milliseconds.	0 – 5237295
explicit (categorical)	Whether or not the track has explicit lyrics (true = yes it does; false = no it does not OR unknown)	--
danceability (numeric)	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.	0 – 0.9850
energy (numeric)	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.	0 – 1
key (integer)	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.	0 – 11
loudness (numeric)	The overall loudness of a track in decibels (dB).	-49.531 – 4.532
mode (integer)	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.	0 – 1

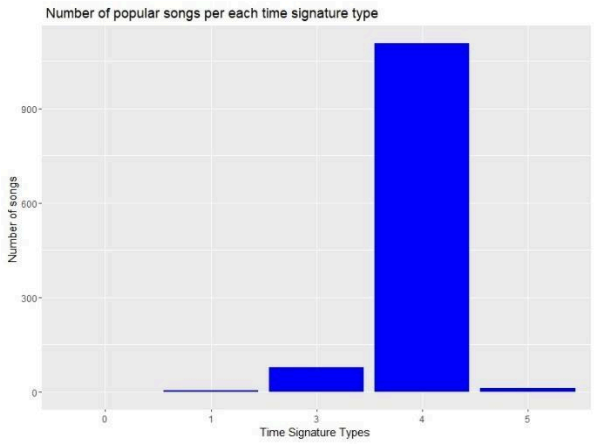
speechiness (numeric)	Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 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.	0 – 0.965
acousticness (numeric)	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.	0 – 0.996
instrumentalness (numeric)	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.	0 – 1
liveness (numeric)	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.	0 – 1
valence (numeric)	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).	0 – 0.995
tempo (numeric)	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.	0 – 243.37
time signature (integer)	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.	0 – 5
track genre (categorical)	The genre in which the track belongs.	--

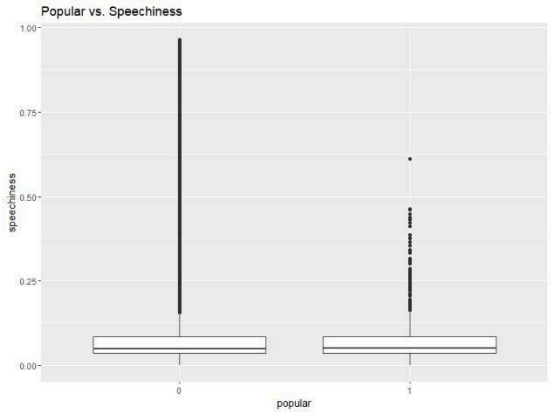
EDA Additional Plots



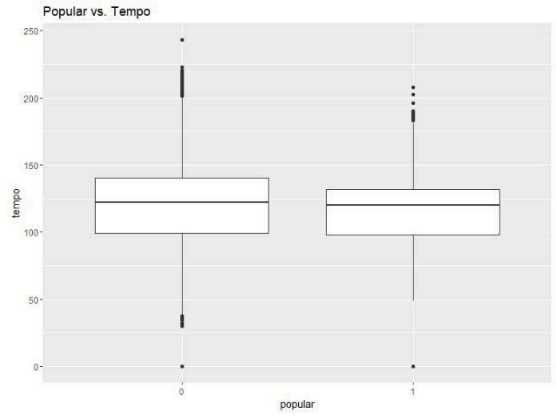




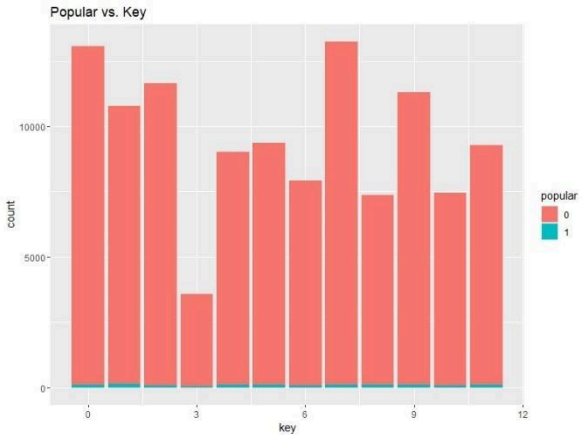




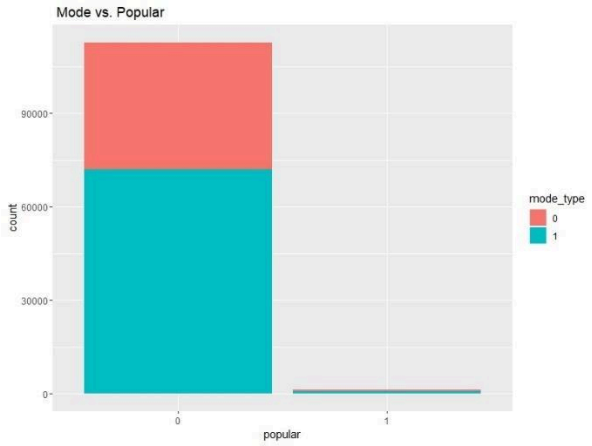
Speechiness does not appear to influence song popularity as non-popular and popular songs have very similar center distributions for this audio feature.



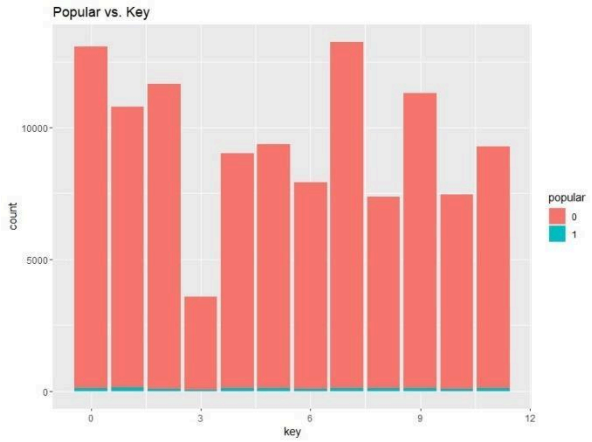
Tempo does not appear to influence song popularity. Both non-popular and popular songs have very similar distributions.



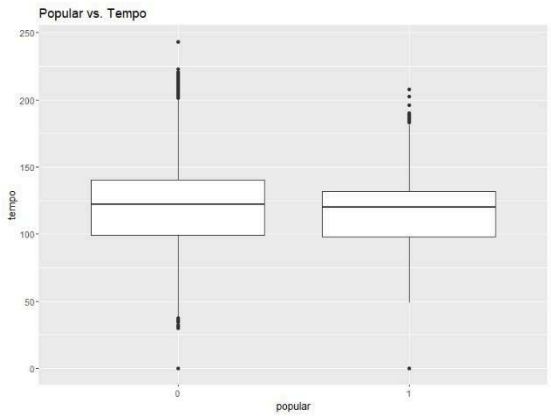
Key does not appear to influence song popularity. It is also difficult to determine if **key** is relevant given that the majority of observations are not popular.



The small percentage of popular songs appear to have a **mode** of 1, which represents the major modality.



Key does not appear to influence song popularity. It is also difficult to determine if **key** is relevant given that the majority of observations are not popular.



Tempo does not appear to influence song popularity. Both non-popular and popular songs have very similar distributions.