Abstract:

This project centers around analyzing song data to create a prediction model for a song's popularity. In doing so, we investigate how some of the most important factors such as energy, danceability, tempo, and more, contribute to a song's popularity - both positively and negatively. In exploring the importance of these aspects on the popularity of songs, we also investigated some trends shown in these features over time through the last few decades.

Our dataset was scraped directly from the Spotify developer console and gives 16 key metrics Spotify gathers on songs, including 'popularity.' In order to determine which features of a song have the greatest impact on its popularity, variable selection strategies such as Lasso regression and stepwise selection were first employed. These techniques returned a list of 6 most influential features in determining a song's popularity - energy, loudness, acousticness, instrumentalness, valence (which describes musical positiveness), and year.

After the feature set was narrowed down, experimentation was done using several different modeling approaches to determine which yielded the best song popularity prediction results. To compare results across different models, cross-validation was used and measures such as mean absolute error, root mean squared error, and R^2 were used. Results were compared for many different modeling approaches including ordinary least squares regression, weighted least squares regression, adding interaction terms, variable transformations with Boxcox, Random Forest, and Catboost. Ultimately, the model that was deemed to have the best combination of predictive accuracy and explainability was an ordinary least squares regression on the 6 most important features and a few interaction terms between them. This model was able to achieve a mean absolute error of 0.137 in predicting a song's popularity on a scale from 0-1, and it had an R^2 of 0.3558.

This project highlighted some important information about which factors are most important in determining a song's popularity, many of which were logical. The one factor that's relation to popularity was not immediately obvious was the 'year' column. However, further analysis into trends in song features like danceability and tempo over time revealed interesting trends in music tastes, which also helped explain why this column was useful in adding prediction power.

Ultimately, this project was a very informative investigation into applying regression techniques to a real-world scenario. It showed not only the many challenges with obtaining a usable dataset, but it demonstrated that models can only be as good as the data backing them, and that one should not just blindly trust models that appear to yield good prediction results without investigation into the reasons they work.

Introduction / Problem Statement:

The goal of this project was to predict the popularity index of a song given associated information such as its danceability rating, energy rating, key, tempo, instrumentalness, time signature, and several more features. This model was trained on some of the songs in our dataset and was then tested by predicting the popularity index of other songs in the dataset. In creating this model, we also tried to investigate the impacts that certain features have on the popularity of a song - identifying which features have the strongest relationship with popularity and how they seem to affect it (positively or negatively).

Data Source:

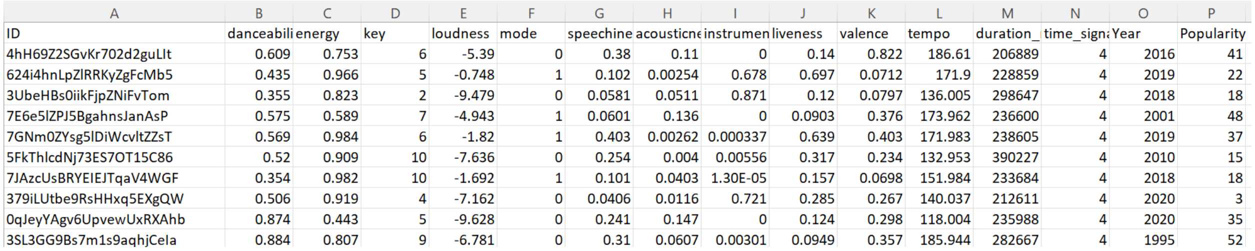
This project presented many unforeseen challenges in obtaining the dataset for our analysis. Our initial approach, which had to be modified later, involved combining data from two different sources. The first dataset was the Spotify songs dataset which can be found on Kaggle at https://www.kaggle.com/mrmorj/dataset-of-songs-in-spotify. This dataset contains song name, genre, time signature, duration, key, and tempo, as well as several metrics created by Spotify including danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, and valence (a measure of a song's positivity).

The second source was the Million Song Dataset from Columbia University located at: http://millionsongdataset.com/. This dataset contains song titles along with other features similar to the first dataset, like loudness, energy, end of fade, tempo etc. but the main feature needed from this dataset is the "hotness" rating - a metric developed for measuring a song's popularity. Some of the other predictors taken from this dataset were 'artist familiarity' and 'artist hotness'.

When we tried to merge the spotify and the million song dataset with song title as key, we discovered that the spotify dataset had many missing song titles. Therefore, we had to scrape the song titles from the spotify website corresponding to those songs. To extract the publishing year and popularity index of the songs hidden inside the Spotify Developer Console, we used https://developers.spotify.com/console.

After the data cleaning, we were able to build a predictive model with the resulting dataset. However, the prediction accuracy seemed suspiciously good. So, when we explored the dataset again from a quantitative point of view, we found that the many of the 'year of publishing' and 'song_hotness' were 0, which caused the model to predict a popularity of zero for many of the input songs, and we found that we could not actually rely on the dataset for these two predictors. Thus, we decided that we would scrape some more data from the spotify website to get the year of publishing for the songs. As for the song_hotness, we found that spotify maintains a secret popularity rating of their songs in their development console. So, we decided to scrape the popularity index of the songs from the development console.

The main dependency on the million song dataset was due to the target variable 'song_hotness'. But as we were able to replace that with the popularity index from the spotify development console, it was warranted that the million song dataset would be removed altogether. The 'artist familiarity' and 'artist hotness' attributes were also removed to predict a song's popularity only on the basis that the model should be based entirely on characteristics of the song and should not rely on outside factors. Moreover, the song titles were removed as the primary key because multiple songs had the same title. As a result, it was not wise to keep the song title as a unique key. Rather we decided to use the Spotify track id as the primary key. All of the fields included in the final dataset are shown below, and the “Popularity” column was the response variable that we hoped to be able to predict.



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
ID	danceabili	energy	key	loudness	mode	speechine	acousticne	instrumen	liveness	valence	tempo	duration_	time_signa	Year	Popularity
4hH69Z2SGvKr702d2guLIt	0.609	0.753	6	-5.39	0	0.38	0.11	0	0.14	0.822	186.61	206889	4	2016	41
624i4hnLpZlRRKyZgFcMb5	0.435	0.966	5	-0.748	1	0.102	0.00254	0.678	0.697	0.0712	171.9	228859	4	2019	22
3UbeHBs0iikFjpZNiFvTom	0.355	0.823	2	-9.479	0	0.0581	0.0511	0.871	0.12	0.0797	136.005	298647	4	2018	18
7E6e5lZPJ5BgahnsJanAsP	0.575	0.589	7	-4.943	1	0.0601	0.136	0	0.0903	0.376	173.962	236600	4	2001	48
7GNm0ZYsg5lDiWcvltZZsT	0.569	0.984	6	-1.82	1	0.403	0.00262	0.000337	0.639	0.403	171.983	238605	4	2019	37
5FkThlcdNj73ES7OT15C86	0.52	0.909	10	-7.636	0	0.254	0.004	0.00556	0.317	0.234	132.953	390227	4	2010	15
7JAzcUsBRYEIEJTqaV4WGF	0.354	0.982	10	-1.692	1	0.101	0.0403	1.30E-05	0.157	0.0698	151.984	233684	4	2018	18
379iLUtbe9RsHHxq5EXgQW	0.506	0.919	4	-7.162	0	0.0406	0.0116	0.721	0.285	0.267	140.037	212611	4	2020	3
0qJeyYAgv6UpvewUxRXAhb	0.874	0.443	5	-9.628	0	0.241	0.147	0	0.124	0.298	118.004	235988	4	2020	35
3SL3GG9Bs7m1s9aqhjCela	0.884	0.807	9	-6.781	0	0.31	0.0607	0.00301	0.0949	0.357	185.944	282667	4	1995	52

Fig 1. Final dataset snapshot

Lastly, we decided to divide the total dataset into training and testing dataset with a 80%/20% split respectively through random sampling.

Methodology:

One of the key things that we tried to maintain is replicate a real-world problem solving approach. In real-life, we hardly get any dataset that is very well defined and structured. In real-life datasets, there remains a lot of missing values, bad values, skewed values, duplicate keys with slightly different predictor values - which can result in poor performance. Data Scientists come up with different strategies to improve the performance of those models. Therefore we decided to work with some difficult datasets and see how we could improve the baseline.

The first question that our project aimed to address was which features are most important in predicting a song’s popularity, and the kind of effect they have on song popularity. This question was addressed first because its results were needed to be able to create the most effective and explainable prediction model possible for song popularity. A variety of different variable selection techniques were used to answer this question, including stepwise selection, forward selection, backward elimination, and LASSO regression.

Lasso regression was the first approach attempted, and it was used because it focuses on reducing the number of features used in a model. It also has the benefit that if two features are highly correlated, it will usually only keep one of them. To maximize the effectiveness of Lasso, we had to first determine the best alpha value to use, which determines the number of variables that ultimately get kept by setting the “budget” allocated for coefficients of different variables. To determine the rough order of magnitude of alpha, we utilized a grid search with scikit-learn GridSearchCV, which builds Lasso regression models with several different choices of alpha values and returns back the alpha value that resulted in the highest prediction accuracy when tested using cross-validation. This helped us determine that the rough order of magnitude of the best alpha value was around 0.1. After this, further experimentation was done with making minor adjustments to the alpha value and evaluating the results using cross-validation by looking at error measures such as mean absolute error, mean squared error, and R^2.

Stepwise selection was also utilized to reduce the feature set and see if it returned the same results as Lasso since stepwise selection uses different methodology for feature selection and does not require the data to be scaled. When limiting the number of variables selected to the same number that were allowed in Lasso, stepwise selection returned the exact same variables in the exact same order of importance that Lasso did, so we could confidently say these were the most influential factors in determining a song's popularity.

After our feature set had been narrowed down to the most important factors in predicting a song's popularity, it was time to address our second question - how can we create a regression model to best estimate the popularity rating of a song? We tried many different approaches to try to create the most accurate popularity prediction model, including Ordinary Least Squares, Weighted Least Squares, adding interaction terms, Box-Cox transformations, Random Forest, and Catboost. Having the smaller feature set was already helpful in that it made it easier to try adding interaction terms to the model and try variable transformations.

To answer this question, we first created a baseline OLS regression model using all the original features from the dataset. We used 80% of the data to train the data and the other 20% to test it to obtain the root mean squared error, mean absolute error, and R^2. Then, an OLS model was created using only the 6 most important features determined from the previous question. We also experimented with adding interaction terms between each of the 6 important features used by multiplying each feature by each other feature and including these in the OLS regression model. The rationale for doing this was that certain effects may be produced by two features coming together, such as high loudness combined with high energy, that may not fully be captured by each individual feature's coefficients.

Boxcox plots were also made for the 6 chosen variables to see if any variable transformations were necessary on the target variable, but the plots showed an optimal lambda value of about 1, which indicated that no variable transformations were necessary. An example of one of the Box-Cox plots can be shown in the Appendix with figure E - though the optimal lambda value seems to be somewhere between 0 and 1, assuming lambda may actually be 0 and doing any log transformations on the target variable only reduced the accuracy of prediction results and was deemed unnecessary.

Additionally, a Weighted Least Squares regression approach was attempted in case the errors on the songs are not independently and identically distributed, though this did not appear to be the case. This was implemented using the iteratively reweighted least squares fitting algorithm discussed in class, which converged quickly but produced a higher MSE and MAE and was thus deemed unnecessary.

As an additional experiment, a random forest regression model was created to reduce variability. This model was created using the scikit-learn RandomForestRegressor package (see bibliography for documentation citation). This approach is very different from any other approaches attempted in that it makes use of decision trees that incorporate some element of randomness in each. This model ended up producing the most accurate results, with a test mean absolute error of 0.1331, but its results are by far the hardest to interpret since random forest regression models provide very little insight into the role each feature plays in determining each song's predicted popularity score. For this reason, this model was not chosen as the final model in favor of using a model with more interpretability.

Another model that was explored was Catboost, a relatively new gradient boosting decision tree model developed by Yandex (see bibliography for documentation citation). As with many other tree-based models, there are a few parameters we can tune such as: minimum number of data points in a leaf, subsample of columns to build each tree, and depth of each tree, etc. A 5-fold cross validation with grid-search process was conducted to try different values for the tuning parameters. The resulting Catboost model with the best set of parameter values was providing a R^2 of 0.45 which is an improvement of the previous best performing linear model. Since the Catboost model's explainability is low and its actual prediction accuracy proved no better than any of the other approaches, it was eventually abandoned in favor of some other models discussed later in the Results section.

Analysis and Results:

First, we carried out Exploratory Data Analysis in order to understand the distribution of our response variable and the correlation between multiple features. We plotted a histogram of the song popularity index as shown below:

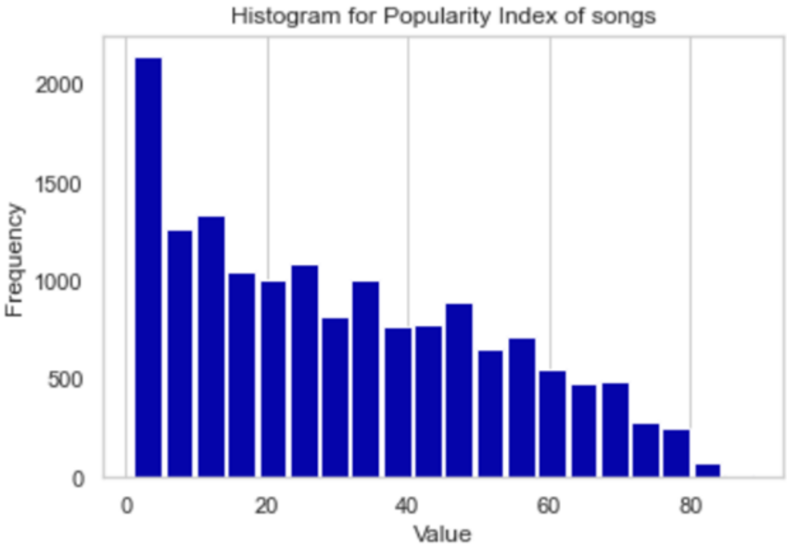


Fig 2. Histogram for popularity Index of songs

As expected, we observed a right skewed distribution of the song popularity index.
The correlation heat map between all variables is plotted below (with only intersections with magnitude of correlation > 0.2 being shown):

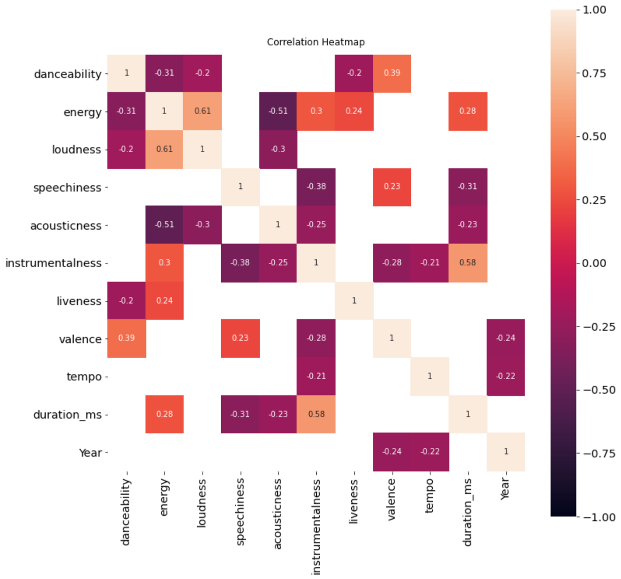


Fig 3. Correlation heat map between variables

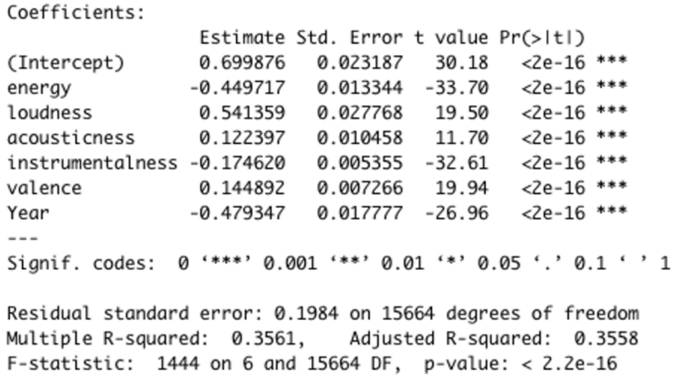
We observed that some independent variables like duration_ms and instrumentalness are highly positively correlated. Instrumentalness is also highly negatively correlated with Popularity (our target variable) which suggests that the final model might probably end up using instrumentalness and excluding duration_ms.

It was also very important to understand if our working dataset had any missing values and if there was, we needed to determine the strategy to deal with them. In this case, our data set did not contain any missing value. Next, we took a look at the unique values of each feature to give us an idea regarding which columns could be considered categorical. We plotted the frequencies of each category for the following variables in the training data: time_signature, mode, and key - having the lowest number of distinct values in the dataset.

From Appendix A, we see that there is a big difference in between the category frequencies of the time_signature variable, i.e. there are a lot more 4’s than 1’s, 3’s and 5’s, also, there are no 2’s. The distribution of the categories between train and test set (Appendix B) were found to be similar, and there were no newer categories in the test data set which could potentially affect our performance after training. From plotting the histograms, we noticed the skewed distributions of the predictor variables in both the training (Appendix C) and test datasets (Appendix D). Again, both train and test dataset numerical variables had very similar distribution, so we could be confident that the patterns learned in train would also be applicable when predicting over the test set.

To answer the question as to which features are most important in determining a song’s popularity, the Lasso and stepwise models for determining the most important features gave us 6 features of the 16 provided that were most influential in predicting popularity: energy, loudness, acousticness, instrumentalness, valence (which describes musical positiveness), and year. It seems logical that many of these features would have a large effect on a song’s popularity, but it did not seem intuitive that the year column was considered important, so we dove deeper into why this might be, and our findings will be presented later in the report. Some experimentation was also done with adding the 7th and 8th most influential factors to the Lasso model, but these showed that adding any features beyond the 6 already identified resulted only in extremely small changes in mean absolute error, mean squared error, and R^2.

The coefficients for each variable returned by the Lasso regression can be seen in Figure 4 below, and they can be used to compare the magnitude of effect of different features on a song's popularity since they use data where all features are scaled from 0 to 1. These coefficients show that loudness has the greatest magnitude effect on the popularity of a song, and the effect is positive - which means that a song with a higher loudness is likely to have a higher popularity. The year column has the second largest magnitude effect on the popularity of a song, and its effect is negative - which suggests that songs made in more recent years are more likely to have a lower popularity score than songs made longer ago. But that being said, we need to keep in mind that songs produced in recent years had less time to generate popularity in Spotify. Energy also has a strong negative effect on popularity, which suggests that songs with high energy are likely to have a lower popularity rating. The rest of the coefficient results shown in the figure can be interpreted in a similar manner.

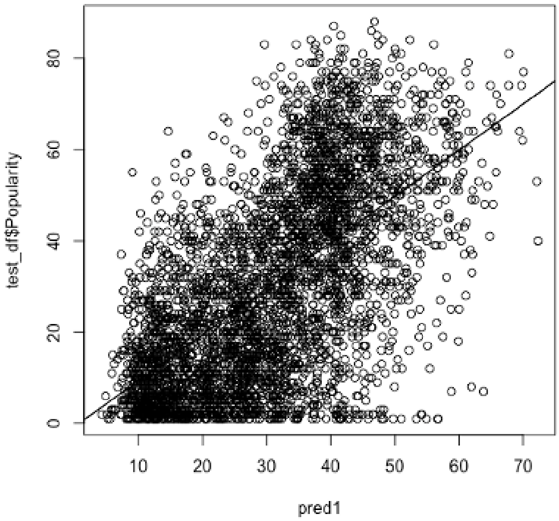


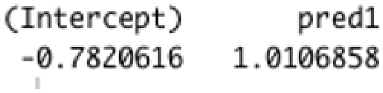
```
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)       0.699876   0.023187   30.18   <2e-16 ***
energy           -0.449717   0.013344  -33.70   <2e-16 ***
loudness          0.541359   0.027768   19.50   <2e-16 ***
acousticness      0.122397   0.010458   11.70   <2e-16 ***
instrumentalness -0.174620   0.005355  -32.61   <2e-16 ***
valence           0.144892   0.007266   19.94   <2e-16 ***
Year             -0.479347   0.017777  -26.96   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1984 on 15664 degrees of freedom
Multiple R-squared:  0.3561,    Adjusted R-squared:  0.3558 
F-statistic:  1444 on 6 and 15664 DF,  p-value: < 2.2e-16
```

Fig 4. Lasso regression output

To determine the best popularity prediction model, the results of much of the experimentation outlined in the proposed methods section can be seen in the table below. Experimentation with all of the methods outlined in the Methodology section were carried out using the ‘leaps’, ‘caret’, and ‘mass’ packages in R (see bibliography for documentation citations). Experiments with interaction terms showed that there were three interaction terms that provided a significant enough increase to prediction accuracy and were deemed worth including in the model, and these terms were energy*loudness, instrumentalness*valence, and year*instrumentalness. The model labeled ‘Final Model’ below was trained to achieve the best combination of prediction accuracy and explainability. It was a simple OLS regression using the 6 most important features as well as the 3 interaction terms discussed above. All of the different approaches that have been discussed were trained on the training set composed of 80% of our data and tested on the other 20% of the data. The error measures of some of the most successful models considered can be seen in the table below. We can see that the unscaled model that contains the small feature set with the interaction terms performed the best, and it had an R^2 of 0.3558 which was about consistent across all the different types of models tried. The plot of this model's predictions against the test dataset's actual results can be seen in figure 5 below. The best fit line of this plot has a slope of almost exactly 1, and a very small intercept of -0.78, which indicates that the model seems to be overestimating and underestimating songs roughly equally and has no clear systematic bias in its predictions.

Baseline model	Small feature set w/o interaction terms	Small (scaled) feature set w/o interaction terms	**Final Model** (small feature set with interaction terms)
RMSE: 0.1995158	RMSE: 0.1713079	RMSE: 0.173929	RMSE: 0.170459
MAE: 0.1606576	MAE: 0.1384511	MAE: 0.1394565	MAE: 0.1374017





```
(Intercept)       pred1
 -0.7820616   1.0106858
```

Fig 5. Fitted vs actual plot

The last linear model trained - and the best performing one - collected the best ideas from the previous linear model but one-hot encoded the key, mode and time_signature variables. Also, a 5-fold cross validation process was carried out to find the best lambda parameter for a Lasso model. The best Lasso model selected 24 out of 30 variables. The selected variables were passed to a final linear regression which outputs an R^2 of 0.368, MAE of 0.137 and MSE of 0.169. The final model summary is shown below. The complete summary including all variable coefficients, standard error, and more can be seen in Figure F in the Appendix.

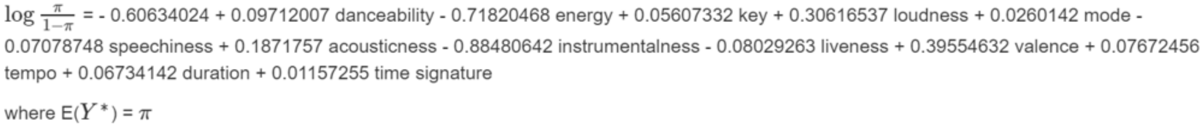
Dep. Variable:	Popularity	**R-squared:**	0.368
Model:	OLS	**Adj. R-squared:**	0.367
Method:	Least Squares	**F-statistic:**	325.8
Date:	Sun, 28 Nov 2021	**Prob (F-statistic):**	0
Time:	17:10:14	**Log-Likelihood:**	-66896
No. Observations:	15671	**AIC:**	1.34E+05

Df Residuals:	15642	**BIC:**	1.34E+05
Df Model:	28		

The next goal of the project was to find the combination of features that would enable music producers/labels/artists to produce songs with a high popularity index. Therefore, we wanted to explore the possibility of creating a classifier to classify between popular and unpopular songs.

We started out by labeling each song in our dataset as popular or not. Due to the right skewed nature of the dataset, we decided to use a popularity index threshold of 35, which is the 60th percentile. We utilized Logistic Regression to create a Classification Model from the training dataset. The model achieved an accuracy of 76% and an F1 score of 0.7124 when run on test data. Upon plotting the ROC curve, we obtained an AUC of 0.76. (Appendix G).

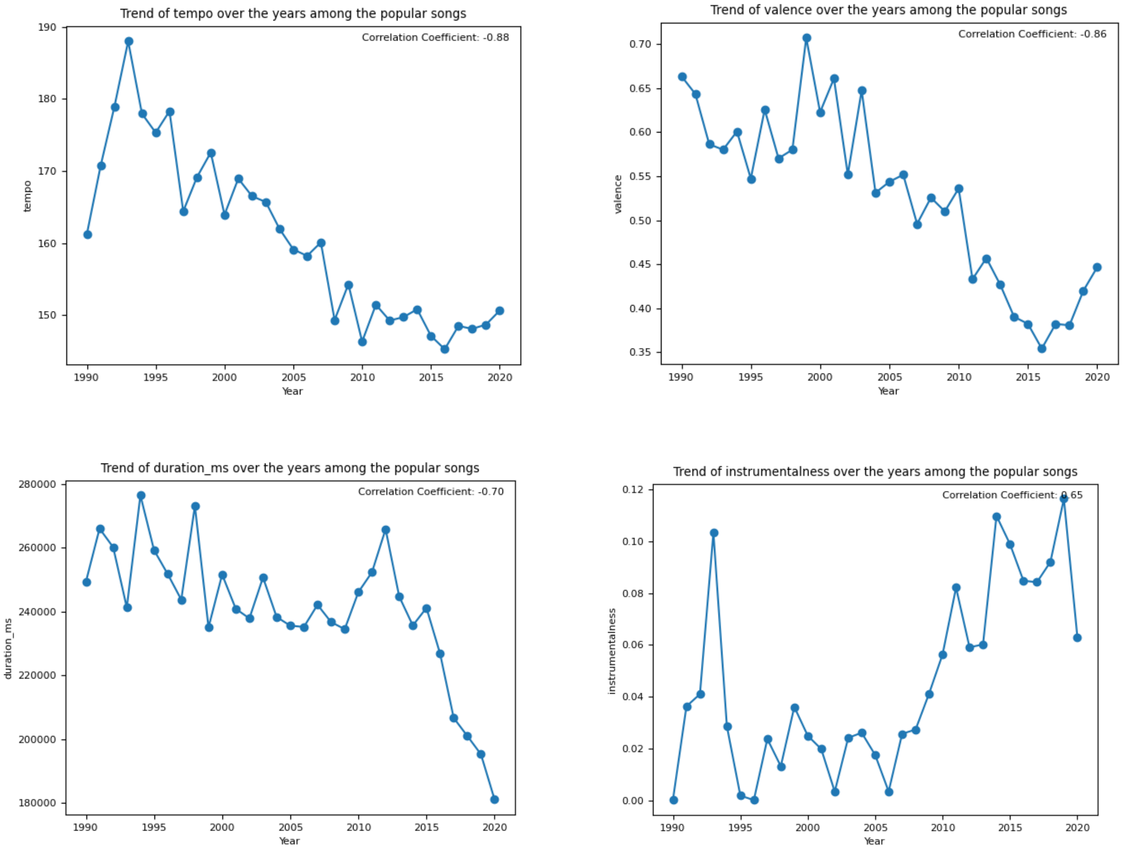
The final equation for the classifier obtained from Logistic Regression is:



$\log \frac{\pi}{1-\pi}$ = - 0.60634024 + 0.09712007 danceability - 0.71820468 energy + 0.05607332 key + 0.30616537 loudness + 0.0260142 mode - 0.07078748 speechiness + 0.1871757 acousticness - 0.88480642 instrumentalness - 0.08029263 liveness + 0.39554632 valence + 0.07672456 tempo + 0.06734142 duration + 0.01157255 time signature

where E(Y^*) = π

The last question that our project aimed to address was how trends of different features in popular songs have changed over time. This question came about partially because the ‘year’ column was found to be one of the most significant features in predicting a song’s popularity. It was suspected that the year column may be highly correlated with several other columns in the dataset, which would mean that including the year column in our prediction model actually provides a lot more information about a song than is initially obvious. This question can be addressed by visualizing the mean of different features of popular songs (those with song popularity above a certain threshold level) over time. Based on that observation, we might be able to find certain trends in those features which can help us to predict the features of the most popular songs in the future. Plots were created to visualize trends in different popular song features over time along with the correlation coefficient between year and each feature in the dataset. Some of the most significant plots are shown below, and the rest can be found in Appendix H.


Trend of tempo over the years among the popular songs

Trend of valence over the years among the popular songs

Trend of duration_ms over the years among the popular songs

Trend of instrumentalness over the years among the popular songs

Fig 6. Time series plots for tempo, valence, duration and instrumentalness

The features which had the strongest correlation with year are tempo, valence, duration, and instrumentalness. Over the years, the most popular songs are found to have, on average, lower and lower tempo. Same goes for the valence of the songs, which is a bad indicator because people are getting inclined towards more depressing or less "positive" themed songs. People may also be getting reluctant to listen to lengthy songs, because on average, the length of popular songs are getting reduced over time. Lastly, the use of instruments seems to be increasing over time. Specifically with the advent of new instruments as well as computer generated artificial tunes, listeners may be more disposed towards instrumentalness. These plots show very interesting trends, and they give insight into why the 'year' column was so useful in predicting a song's popularity. Since a song's year can be so highly correlated with other features such as its tempo, valence, and duration, including the year gives some insight into these features without needing to explicitly include these features in the prediction model.

Conclusions:

From our analysis, we observed that an Ordinary Least Squares model with one-hot encoded terms and feature interaction terms produced the best combination of performance and explainability. From the OLS regression model, we found that the most important features influencing a song's popularity are instrumentalness, energy, valence, loudness, acousticness and year. Since year is a factor that an artist or a music producer cannot manipulate, we observed the trends of the different features over time and found strong correlations with tempo, valence, song duration and instrumentalness. This helps explain why year was such an important factor in the model, as it inherently gives clues about several other features.

Finally, it is to be noted that a model can only be as good as the data it is being trained on. For all the models that we have analyzed in this project, the coefficient of determination has been found to be rather low which indicates that the models do not explain the variation in the response variable by a lot. We have not been able to derive a clear pattern in the popularity index from the predictor variables. Additionally there are multiple external factors coming into play in determining the popularity of a song. A song may go viral by word of mouth or due to its eccentric nature, a number by a popular artist can be extremely popular even if it does not have the attributes determined by our model. For this reason, it is not expected to be possible to create a model that extremely accurately predicts the popularity of a song, but our model helps provide a lot of insight into some influential factors.

For future research work, we can divide the dataset based on genres and analyze the trends and impact of different features on the response variable for each genre. This might also allow us to create a more accurate popularity prediction model as popular songs within a certain genre might be more likely to show similarities in some of the features in the dataset.

Lessons Learned

One of the most important lessons learned from this project and throughout this class has been not to blindly apply modeling and regression techniques when they don’t make sense. This problem was especially highlighted when we initially had a dataset that contained many 0 values for song popularity ratings. Though we were able to create a surprisingly accurate popularity prediction model for this data, we found that this model relied most heavily on factors such as time at which fade out begins or tempo. It didn’t make much sense logically that these factors should be the most important in determining a song’s popularity, and digging deeper into this issue revealed that they were only appearing to be important due to the fact that these values being 0 were usually an indication that song popularity was also 0. This taught us a valuable lesson in not just blindly trusting our model, and showed us that we must pay attention to ensure that its implications are actually logical.

Another important lesson was that it is not always going to be possible to create a model that explains a large amount of variance in your dataset. Some datasets do not necessarily lend themselves well to prediction models as there are too many external factors affecting the response variable. In this situation, we learned that it is better to explain as much of the variance as possible while keeping the models understandable to derive the most insight possible from the model. While it may be tempting to use a more complex machine learning approach to achieve a marginally better R^2 value, we found that it is not always worthwhile to obsess over measures like R^2 alone. Instead, it is important to consider whether the modeling approach being used makes sense given the context of the problem and what we know about it in real life. In other words, a model with a higher R^2 is not always objectively a better model, and we must use our own judgement when deciding what modeling approach works best instead of only considering error measures.

Appendix:

Exploratory Data Analysis Figures

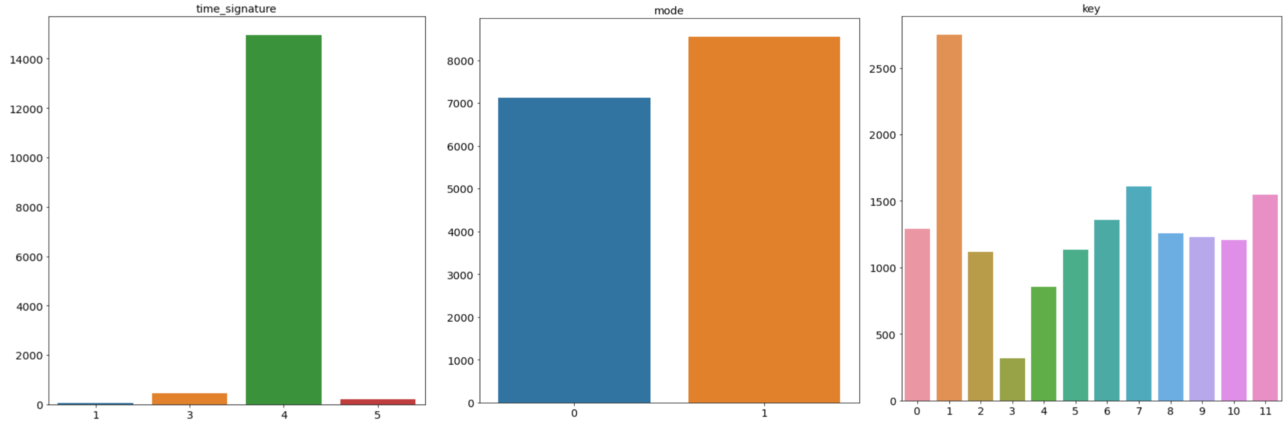


Fig A. Histogram for categorical variables from Training data

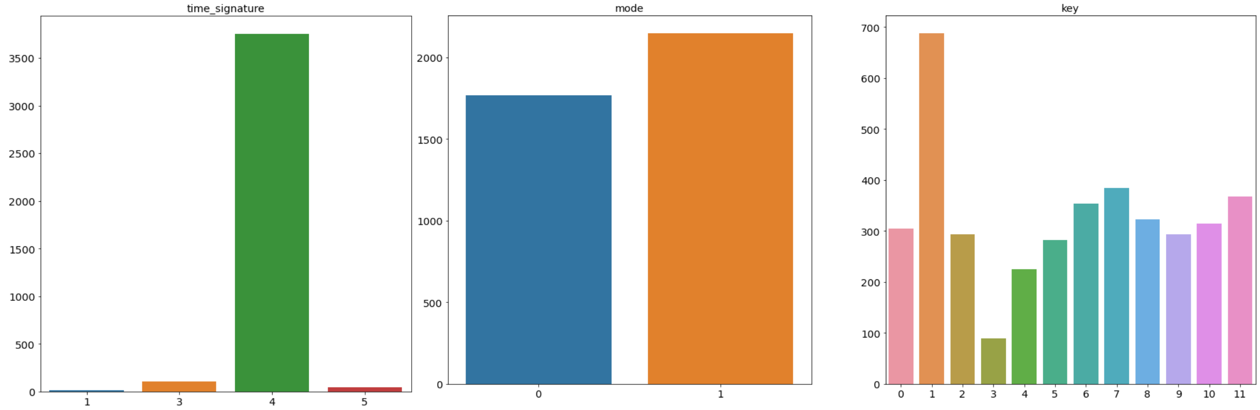


Fig B. Histogram for categorical variables from Test data

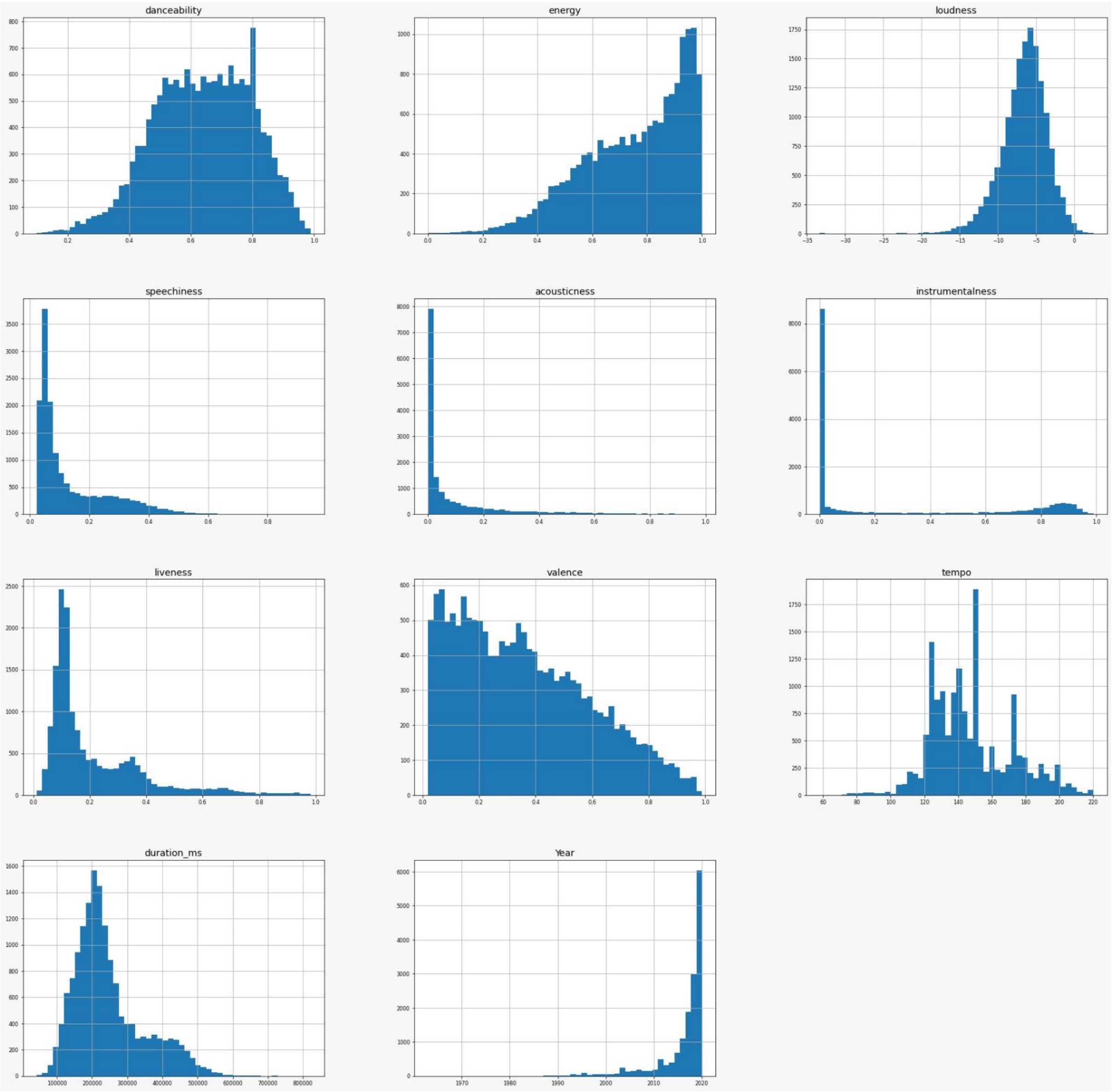


Fig C. Histogram for numeric variables from Test data

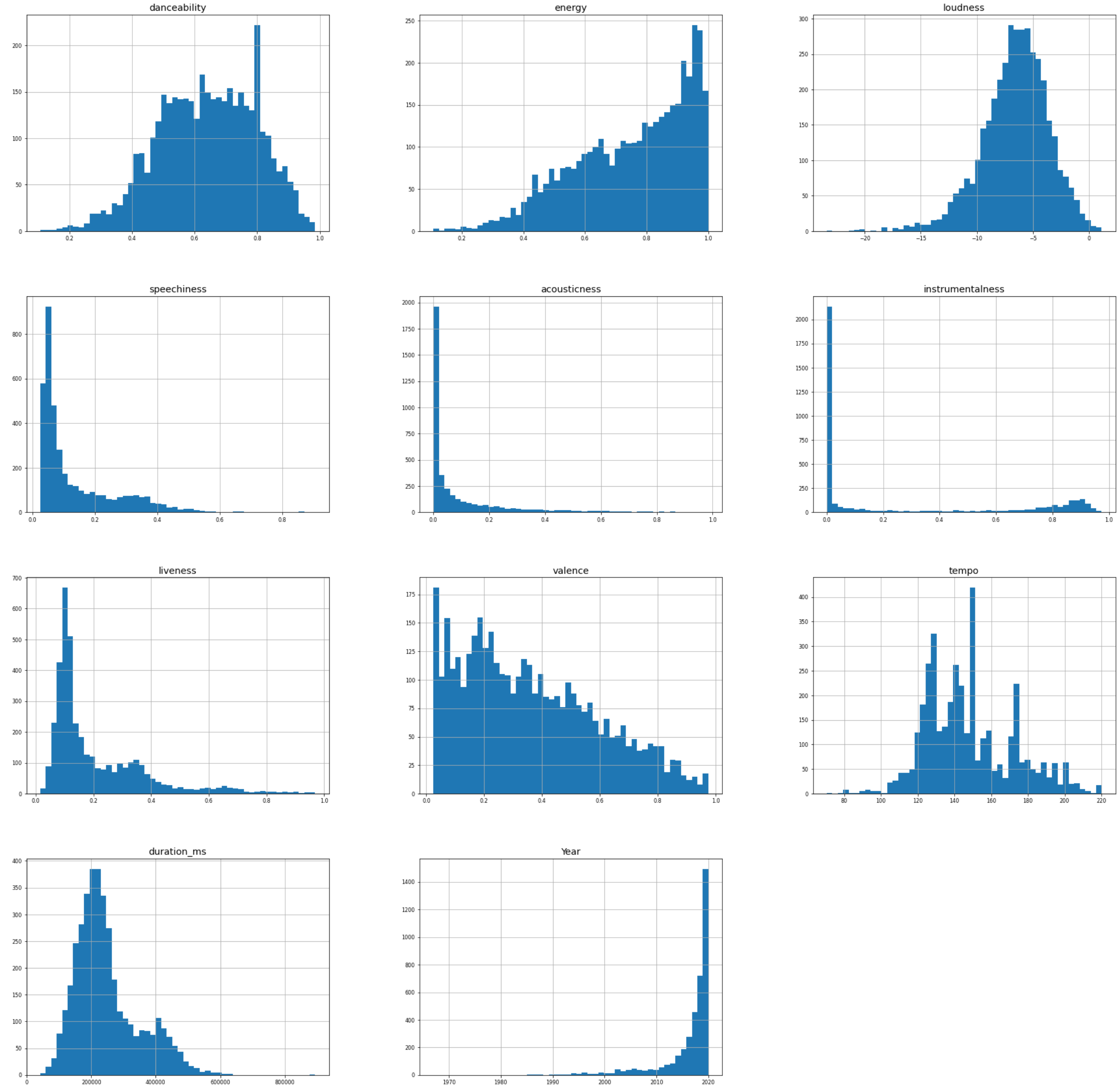


Fig D. Histogram for numeric variables from Test data

Predictive Model Building

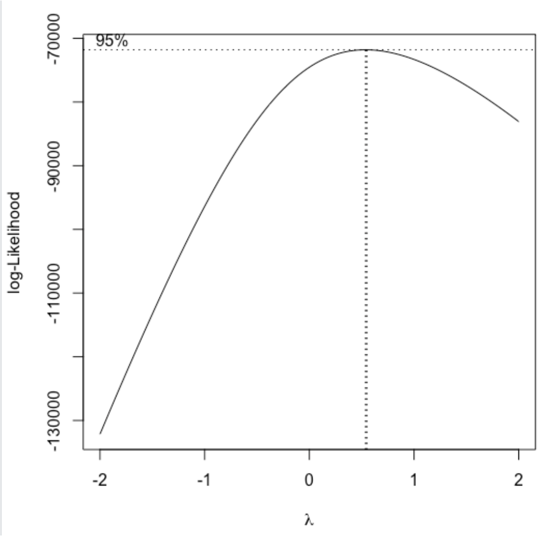


Fig E. Box-Cox Plot for Popularity against all predictors

	coef	std err	t	P>|t|	[0.025	0.975]
const	1181.5698	44.425	26.597	0	1094.491	1268.648
danceability	7.2369	1.122	6.452	0	5.038	9.435
energy	-50.5533	2.024	-24.98	0	-54.52	-46.587
loudness	2.8028	0.164	17.069	0	2.481	3.125
speechiness	-4.2287	1.24	-3.411	0.001	-6.659	-1.799
acousticness	12.8004	0.942	13.59	0	10.954	14.647
instrumentalness	-12.6351	0.747	-16.905	0	-14.1	-11.17
liveness	-4.0945	0.843	-4.858	0	-5.747	-2.442
valence	13.1222	0.878	14.939	0	11.4	14.844
duration_ms	-4.19E-06	1.80E-06	-2.326	0.02	-7.73E-06	-6.60E-07
Year	-0.7437	0.029	-25.465	0	-0.801	-0.686
key_0	98.4378	3.734	26.361	0	91.118	105.757
key_1	97.43	3.727	26.141	0	90.125	104.736
key_2	98.6127	3.734	26.412	0	91.295	105.931
key_3	99.5321	3.807	26.148	0	92.071	106.993
key_4	99.3549	3.73	26.638	0	92.044	106.666

key_5	98.132	3.738	26.252	0	90.805	105.459
key_6	97.7009	3.732	26.182	0	90.386	105.015
key_7	98.4899	3.727	26.425	0	91.184	105.796
key_8	98.5281	3.737	26.362	0	91.202	105.854
key_9	97.7426	3.737	26.153	0	90.417	105.068
key_10	99.2265	3.733	26.581	0	91.91	106.544
key_11	98.3823	3.728	26.393	0	91.076	105.689
time_signature_1	295.5749	11.252	26.269	0	273.52	317.63
time_signature_3	295.0828	11.153	26.458	0	273.222	316.943
time_signature_4	294.5613	11.121	26.487	0	272.763	316.36
time_signature_5	296.3509	11.162	26.55	0	274.472	318.23
energy_loudness	-2.0424	0.211	-9.693	0	-2.455	-1.629
instrumentalness_valence	-11.7109	1.735	-6.749	0	-15.112	-8.309

Fig F. Final Linear Model Summary

Logistic Regression:

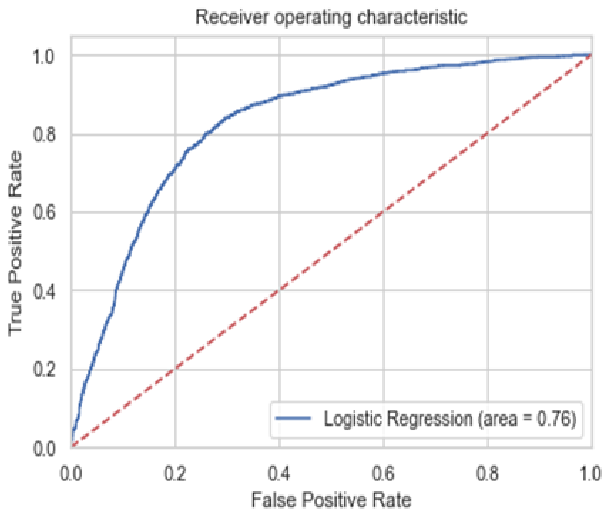


Fig G. ROC curve for Logistic Regression

*Code reference for all modeling code: https://github.com/willemt25/isye6414-group7.git
*For scraping Spotify song popularity:
https://colab.research.google.com/drive/1zQDkefUhGh-UeerchbuKdr_gKwZWRfNj
*For scraping Spotify song year:

https://colab.research.google.com/drive/1s0tUFIyubO1JUvJMzybjE6P3fIQOwGt1?usp=sharing
*For scraping Spotify song title:
https://colab.research.google.com/drive/1WHNt6DO_3FhD5TtvOzXcI-V0QBuPxEuT

Trends Over Time:

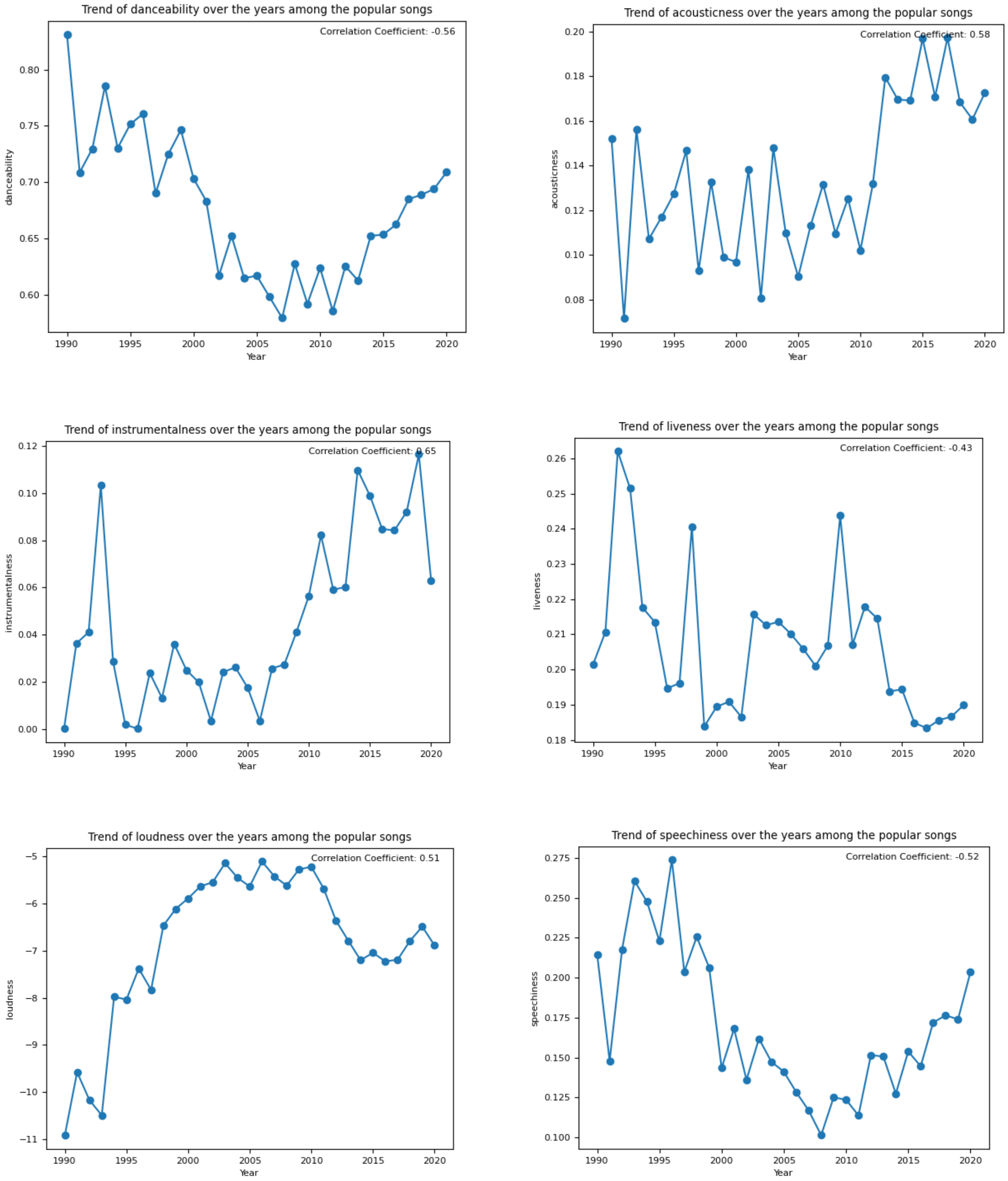


Fig H. Time series plots for song features

Bibliography and Credits:

A Short Introduction to the caret Package. (n.d.). Retrieved from https://cran.r-project.org/web/packages/caret/vignettes/caret.html

CatBoostRegressor. (n.d.). Retrieved from https://catboost.ai/en/docs/concepts/python-reference_catboostregressor

Package leaps. (n.d.). Retrieved from https://cran.r-project.org/web/packages/leaps/index.html

Ripley, B. (2021, May 03). Support Functions and Datasets for Venables and Ripleys MASS [R package MASS version 7.3-54]. Retrieved from https://cran.r-project.org/web/packages/MASS/index.html

Sklearn.ensemble.RandomForestRegressor. (n.d.). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

Sklearn.model_selection.GridSearchCV. (n.d.). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html