Analysis of Top 100 Songs on Spotify, 2010-2019

Group 6: Anchal Lamba, Stephen Fujimoto, Jungwon Yoon

January 24, 2023

1 Introduction

Spotify has grown to be one of the largest music streaming services in the world, with over 422 million users and 82 million tracks since its debut in 2008. Due to its increasing popularity and relevance today, we want to explore which songs are trending on the platform and analyze trendy songs' features. Also, we would like to comprehend global users' listening behaviors and preferences, and how trends in music have changed by each year. Furthermore, such an analysis could potentially provide Spotify a better understanding of trends and recommendations for their target audience by type of music, genre, and artist.

Our main problems we wanted to tackle using this data set were: 1) predicting the level of popularity of a song based on its features and 2) classify the genre of the a song based on its features..

1.1 Dataset

This project utilized a dataset available publicly on Kaggle (can be found here) and containing the top 100 songs on Spotify from the years 2010 to 2019. The dataset has a total of 1000 rows, 100 rows for each year, and 17 columns. Each song is described with its title, artist, genre, year released, added (day the song was added to Spotify's Top Hits playlist), bpm (beats per minute), nrgy (measure of how energetic the song is), dnce (ease of dancing to the song), dB (loudness), live (the probability the song is a live recording), val (positivity of the mood), dur (song's duration/length), acous (measure of how acoustic the song is), spch (measure of how focused the song is on spoken word), pop (song's popularity score), top year (year the song was in a top hit list), and artist type (solo, duo, trio, or a band).

2 Mathematical Notations and Proposed Methods

We utilized these statistical methods we learned in the class to analyze our data and extract meaningful insights and (2) optimize our code runtime and find the most impactful and model to predict trendy songs.

2.1 Linear Regression

Multiple linear regression was used to model the relationship between a song's characteristics and a continuous variable, such as the song's popularity score (pop). The method we used was least-squares, which involves estimating the regression coefficient by minimizing RSS (residual sum of squares), where

$$RSS[\beta] = (y - X\beta)^T(y - X\beta)$$

Finding the estimate of β becomes a problem of solving the linear system $X^TX\beta = X^Ty$. Therefore, $\hat{\beta} = (X^TX)^{-1}X^Ty$

2.2 LASSO Regression

LASSO is a regression analysis method that utilizes an L1 penalty to regularize and achieve a sparser solution [2].

The goal of LASSO is to minimize the following:

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}B_j)^2 + \alpha\sum_{j=1}^{p}|B_j|$$

α is the penalization parameter that shrinks the correlations of insignificant predictors to 0, resulting in feature selection for sparser models. This can also be written as:

$$RSS + \alpha\sum_{j=1}^{p}|B_j|$$

where RSS is residual sum of squares, represented by $\sum_{i=1}^{n}(y_i - \hat{y_i})^2$.

2.3 Singular Value Decomposition (SVD)

We built this method to classify types of the songs and group them based on their features like beats and measure of energy.

1. Principal Component Analysis (PCA)
   (a) PCA is a method to decompose high-dimensional data using Singular Value Decomposition
   $$X = U\Sigma V^T$$
   Where we standardize X centered at mean 0.
   (b) We considered 9 numerical columns in our dataset, We wanted to remove dimensions that do not add much information to our data
   (c) Used this preprocessing method (reduced dataset) for our classification that will be introduced in the next section.

2.4 Classification Algorithms

We implemented logistic regression and XGBoost to classify songs into genres based on numerical features such as "bpm", "dnce", etc.

1. Logistic Regression (Multinomial)
   (a) Used on multiclass classification using cross-entropy loss, which is fitting across the entire probability distribution
   (b) Follows a linear predictor function to predict the probability that observation i has outcome k (type of genre of a song) [3]:

$$f(k,i) = \beta_{0,k} + \beta_{1,k}x_{1,i} + \beta_{2,k}x_{2,i}...$$

   (c) Outputs our outcome by

$$Pr(Y_i = 1) = \frac{e^{\beta_1 X_i}}{\sum_{k=1}^{K} e^{\beta_k X_i}}$$

   In our case, $Y_i = 0, 1, 2$

2. XGBoost
   (a) Used the Gradient Tree Boosting technique. This algorithm is a gradient boosting used with decision trees method descend the gradient of a differentiable loss function like Gradient Descent, yet Boosting technique does it by introducing a function instead of finding a set of parameter.
   (b) Gradient Descent repeats following [4]:

$$\theta^{(t)} = \theta^{(t-1)} - \alpha f'(\theta^{(t-1)})$$

   (c) While Gradient Tree Boosting repeats following [5]:

$$F_t(x) = F_{t-1}(x) + \gamma_t h_t(x)$$

2.5 Parallel Computing

We also used the multiprocessing package in Python to improve code run time, specifically for plotting functions during exploratory data analysis (EDA).

3 Data Cleanup

Before we proceeded with any analysis, we chose to make three major changes to the dataset:

1. Remove any rows with NA values

2. Top genre: There was a significant difference between the number of dance pop songs (357) and other genre categories. The next largest category was 57. Using parallel computing, we created more evenly divided genres to better represent each category in our analysis.

3. Top artist: There were 444 unique artists in our dataset, so we narrowed our search to those who appeared on the top 100 songs list at least 10 times. Specifically, we focused on: Taylor Swift, Drake, Calvin Harris, Rihanna, Ariana Grande, Bruno Mars, Maroon 5, Post Malone, Jason Derulo, Ed Sheeran, and Chris Brown

4 Exploratory Data Analysis

To better understand the nature of our dataset, we pursued exploratory data analysis (EDA) - exploring and visualizing various trends with line and bar graphs, histograms, box plots, and correlation matrices. Our analysis was tailored to the following questions:

1. Are there significant correlations between characteristics of a top song? What are the highly correlated predictors of popularity scores?

2. Do certain song genres, top artists, and artist types have higher or lower popularity scores?

3. Do certain years in which songs were a hit contain different distributions in song characteristics than other years?

4. Are there significant correlations between characteristics of a top song? What are the highly correlated predictors of popularity scores?

4.1 Question 1

We used a correlation matrix to identify significant correlations among numerical predictors of our dataset.

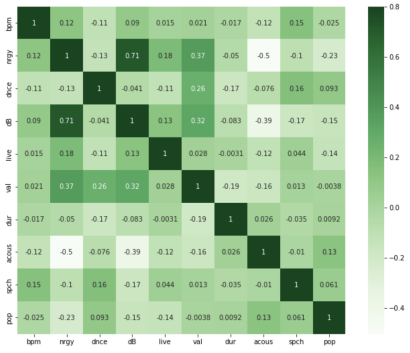

Figure 1: Correlation Matrix

We found that there was a high positive correlation of 0.71 between a song's energy level (nrgy), and how loud a song is (dB). We also found a negative correlation of 0.5 between a song's energy level (nrgy) and how acoustic it is (acous). There were also decent positive correlations between:

1. How positive the mood of a song was (val) and its energy level (nrgy) of 0.37
2. How positive the mood of a song was (val) and its loudness (dB) of 0.32
3. How positive the mood of a song was (val) and its danceability (dnce) of 0.26

Lastly, there was a decent negative correlation of 0.39 between how acoustic a song is (acous) and its loudness (dB).

Additionally, the following output contains the correlations between the song characteristics and its popularity score:

```
nrgy     0.234011
dB       0.145403
live     0.137305
acous    0.128195
dnce     0.093176
spch     0.061441
bpm      0.025058
dur      0.009219
val      0.003752
```

Figure 2: Output of correlations in descending order

While a song's energy level has the highest correlation to its popularity score ("pop"), the value is still quite low, just as the rest of the correlations are. We will have to take this into account when performing linear regression and analyzing our model.

4.2 Question 2

To answer this question, we visualized box plot distributions of popularity scores by genre, top artist and artist type:

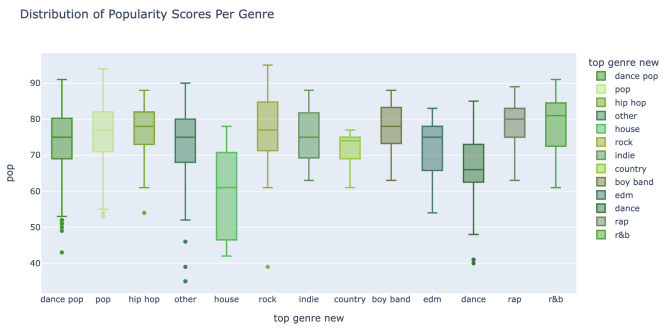

Figure 3: Box plot distribution of popularity scores by genre

House music has significantly lower popularity scores, as its box barely overlaps with any others. Dance pop, pop, hip hop, other, and rock genres all have outliers below the lower quartile, meaning that there were a few songs that had significantly lower popularity scores than the average of that genre.

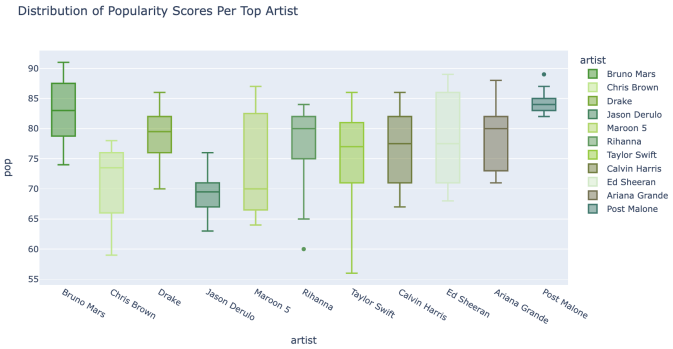

Figure 4: Box plot distribution of popularity scores by top artist

Taylor Swift has the largest range, meaning that at least one of her songs had a much smaller popularity score than her average. Bruno Mars and Post Malone have relatively higher popularity scores, while Chris Brown and Jason Derulo have relatively lower popularity scores.

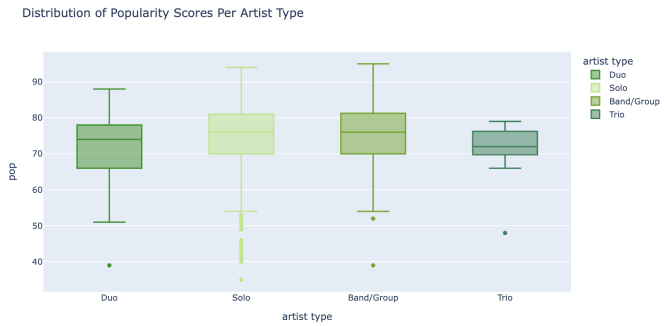

Figure 5: Box plot distribution of popularity scores by artist type

Popularity scores are similarly distributed among all artist types, as the box plots overlap one another. Also, solo artists have many more outliers below the lower quartile, meaning that there were a few songs that had significantly lower popularity scores than the average of that artist type.

4.3 Question 3

We created line graphs for each characteristic and the changes in average values from 2010 to 2019.

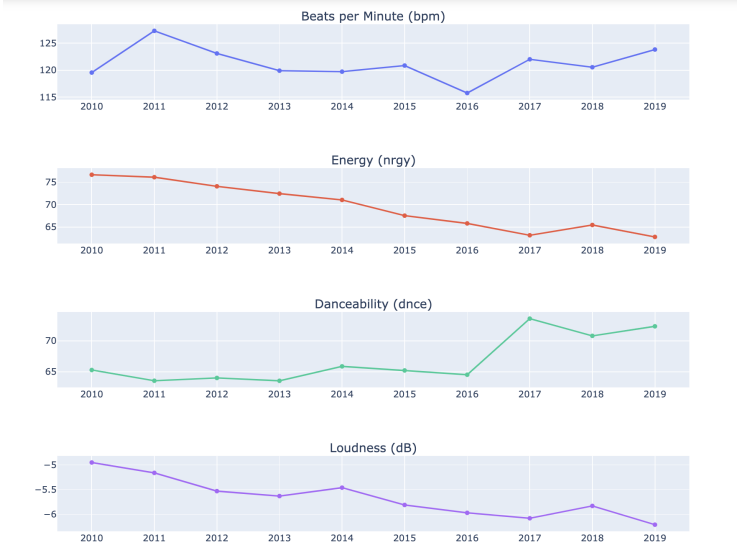

Figure 6: Line graphs for bpm, nrgy, dnce, and dB

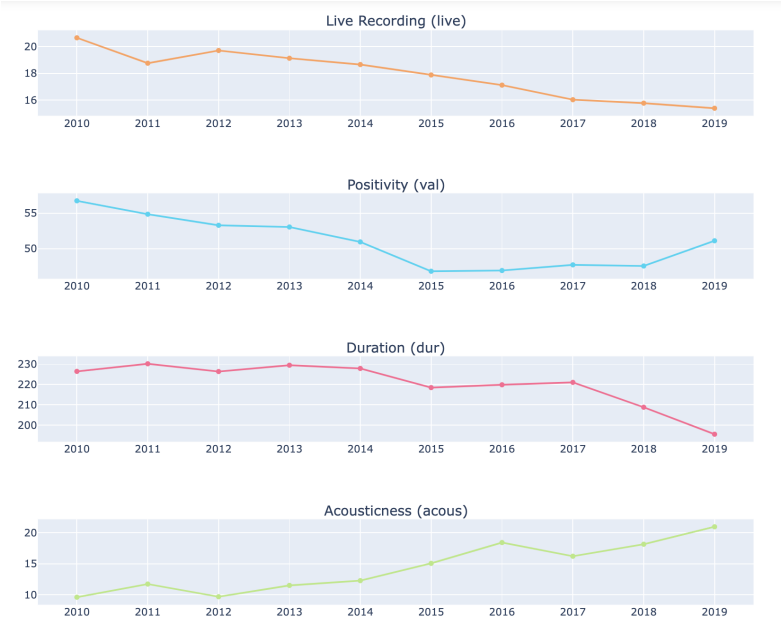

Figure 7: Line graphs for live, val, dur, and acous

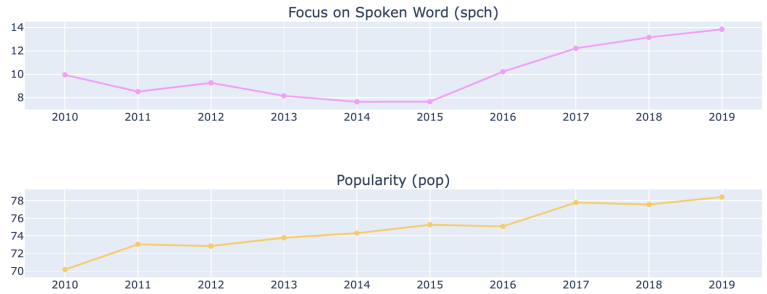

Figure 8: Line graphs for spch and pop

We found that a song's energy level ("nrgy"), loudness ("dB"), duration ("dur"), positivity ("val"), and the likelihood of a song being a live recording ("live") have general downward trends since 2010. On the other hand, a song's popularity (pop) and how acoustic ("acous") it is has general upward trends. Lastly, the song's focus on a spoken word ("spch"), its beats per minute (bpm), and danceability ("dnce") rates have generally upward trends since 2016.

5 Data Analysis

We analyzed our data in depth using proposed methods explained above.

5.1 Linear Regression

Linear regression is fundamental for predictive analysis, modeling the relationship between one or more independent variables and a numerical dependent variable. For our dataset, we chose to explore the relationships between the song characteristics and its popularity score (pop).

5.1.1 Model Diagnostics

After data cleaning and visualization, it was imperative that we check the linear regression assumptions of normality, equal variance, and independence. If one of these assumptions is violated, we cannot proceed with any modeling for our data analysis. The next steps would be to re-evaluate the dataset and examine any missing outliers skewing our data excessively and/or make transformations to the response variable.

As a first step, we checked the normality assumption for our target variable "pop" score. By creating a normality probability plot on the interaction term model, we were able to observe that the points did not follow the line at all, with large quantiles (positive and negative) indicating heavy tails in the data's distribution. Hence, we concluded that the data violated the normality assumption.

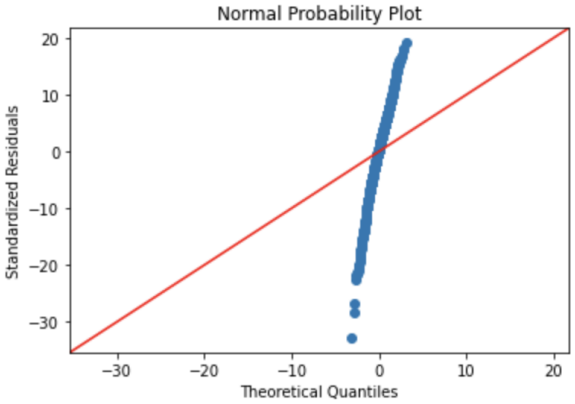

Figure 9: Normal probability plot of the pop variable, before transformation

In order to fix the problem, we then proceeded with the Box-Cox transformation method, which finds the optimal lambda value to transform our pop values to be distributed normally.

Using the scikit-learn PowerTransformer package [1], we found our optimal lambda value was 3.27087407. After transformation, we observed the points falling into a relatively straight line on the plot.

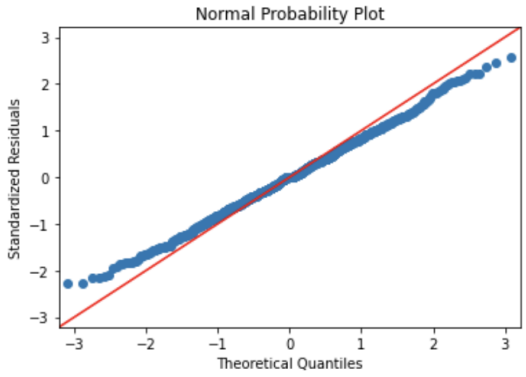

Figure 10: Normal probability plot of the pop variable, after transformation

We also performed the Shapiro-Wilks test to quantify the ability to accept/reject the normality assumption. With our p value of approximately 0.4178 being greater than = 0.05, we failed to reject the null hypothesis, concluding that that the normality assumption holds on transformed values.

To visualize our data's compliance to the homoscedasticity assumption, we also evaluated the Residuals vs. Fitted plot, as well as the Scale-Location plot.

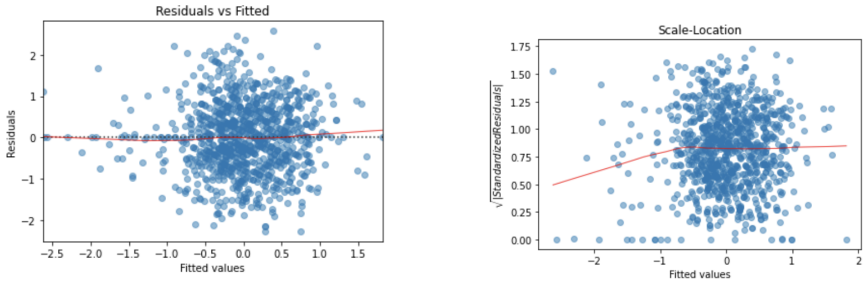

Figure 11: Homoscedasticity Plots

The "Residuals vs Fitted" graph has a relatively straight horizontal line around zero, so the linearity assumption holds pretty well. The graph has a scale of 2, and the points are randomly and evenly spread out around the line. In the "Scale-Location" graph, the points are randomly and evenly spread out around the red line. The only concern is that the line is slightly bent from the left side. Regardless, the assumption for equal variance holds decently under the observations from these two plots.

We also performed the Levene's test to quantify the ability to accept/reject the equal variance assumption among categorical variables. After checking if (1) top years, (2) top genres, and (3) artist types have equal variances in song popularity scores, we concluded that the homoscedasticity assumption was held, and proceeded with regression model building.

5.1.2 Linear Regression

After selecting the variables, we used LU decomposition and Cholesky decomposition to find the least-squares solution of $X^T Xb = y$ to find the regression coefficients and their standard errors. The estimates are displayed below:

Variable	Regression Coefficients	Standard Error
intercept	$1.575 * 10^{-1}$	0.4667
bpm	$6.959 * 10^{-5}$	0.0012
nrgy	$-1.319 * 10^{-2}$	0.0031
dnce	$-1.913 * 10^{-4}$	0.0027
dB	$2.189 * 10^{-2}$	0.0217
live	$-4.569 * 10^{-3}$	0.0023
val	$4.872 * 10^{-3}$	0.0016
dur	$1.479 * 10^{-3}$	0.0008
acous	$9.212 * 10^{-4}$	0.0018
spch	$1.288 * 10^{-3}$	0.0035
top year	$7.464 * 10^{-2}$	0.0117
artist type	$-7.495 * 10^{-2}$	0.0406
top genre broad	$2.594 * 10^{-2}$	0.0097

5.2 LASSO Regression

Given the current regression coefficients and their standard errors, we wanted to explore if a sparser model would improve these estimates, as well as run time. We chose LASSO regression to achieve this task.

To visualize the optimal α value, we plotted the LASSO coefficients versus the regularization parameter, α. We can observe that initially, the model contains high magnitudes of predictor coefficients. As α (regularization parameter) values increase, the coefficient estimates converge to approximately zero.

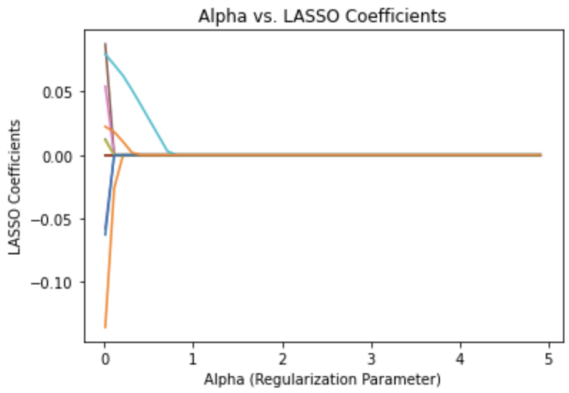

Figure 12: Plot of alpha values against LASSO coefficients

After using the K-Fold method as a means of cross validation, we found that the optimal alpha value for the model is about 0.0158. We then re-trained and tested the LASSO model with our optimal α value to find that the mean squared error was around 0.9615, r^2 for the training set was 11.25, and r^2 for the testing set was 13.05. With such high MSE and low accuracy scores, our model did not perform well with our data; we will have to take this into consideration when modeling the selected features from this test. We still received the LASSO coefficients for each predictor, and they are outputted below:

	Predictor	Coef
0	bpm	0.0
1	nrgy	-0.130325
2	dnce	-0.0
3	dB	0.0
4	live	-0.053694
5	val	0.078155
6	dur	0.046613
7	acous	0.008594
8	spch	0.005281
9	top year	0.078759
10	artist type	-0.05207
11	top genre new	0.02236

Figure 13: Table of predictors and their LASSO coefficients

The song characteristics beats per minute ("bpm"), danceability ("dnce"), and loudness ("dB") have correlations of 0, meaning that they are not significant enough to keep in our model.

Lastly, we plotted the fitted values with the residuals from our model's predictions.

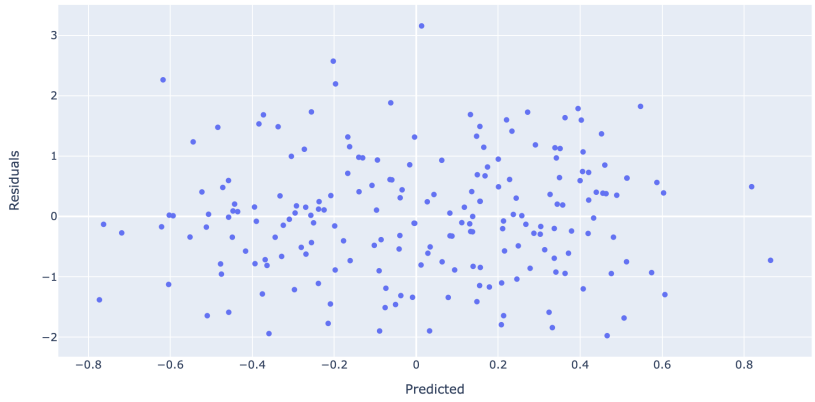

Figure 14: Plot of Residials vs. Fitted Values

The data points are randomly and evenly scattered about the horizontal line y = 0, so the linearity assumption holds pretty well. Also, the residuals roughly form a horizontal band around the line, indicating that the equal variance assumption holds as well.

We re-ran LU decomposition and Cholesky decomposition with the selected features from LASSO. The estimates of the regression coefficients and their standard errors are outputted below:

Variable	Regression Coefficients	Standard Error
intercept	-0.0113	0.0024
nrgy	-0.0.997	0.3006
live	-0.0045	0.0023
val	0.0050	0.0015
dur	0.0015	0.0008
acous	0.0008	0.0018
spch	0.0007	0.0.0033
top year	0.0754	0.0115
artist type	-0.0714	0.0398
top genre broad	0.0250	0.0096

After LASSO regression, the estimate of the variance decreased slightly from

about 0.8913 to about 0.8888. LASSO also cut down on the number of predictor variables providing a more simple model, which improved computational time as there are less variables involved. We looked at the runtime to compute the coefficients and standard errors using LU and Cholesky as well as using the pre-LASSO variables and post-LASSO variables. We found that LU decomposition using the post-LASSO variables was the fastest (32.3 μs), followed by LU decomposition using the pre-LASSO variables (33.7 μs), then Cholesky using post-LASSO variables (169 μs), and finally Cholesky using post-LASSO variables (187 μs). Thus, due to these small improvements, we chose the second model as our final predictive model for popularity score.

5.3 Principal Component Analysis (PCA)

The second problem this project tacked was attempting to classify the genre of a song based on its features by reducing the dimensionality of the dataset via Principal Component Analysis (PCA). (Discussion using PCA as a preprocessing algorithm for other learning algorithms is provided in the next section.) To do so, we considered nine numerical variables ("bpm", "nrgy", "dnce", "dB", "live", "val", "dur", "acous", "spch"). We specifically focused on three genres ("dance pop", "hip hop", "rap"). This is because when we cleaned up our dataset, we merged all types of pop genres except dance pop into a general "pop" genre. The general "pop" genre was excluded from our model then as we assume the genre was to broad for effective classification. After broadening the genre categorization, we were left with thirteen genres to potentially classify into. However, since our dataset had genres with very few entries (such as having the EDM genre having 17 songs out of a total of 1000 songs), we figured that we would not have enough information to classify smaller genres and therefore excluded them. After subsetting the data in this way, we ran our PCA on 487 rows and standardized our numerical variables onto a unit scale, with mean equal to 0 and variance equal to 1.

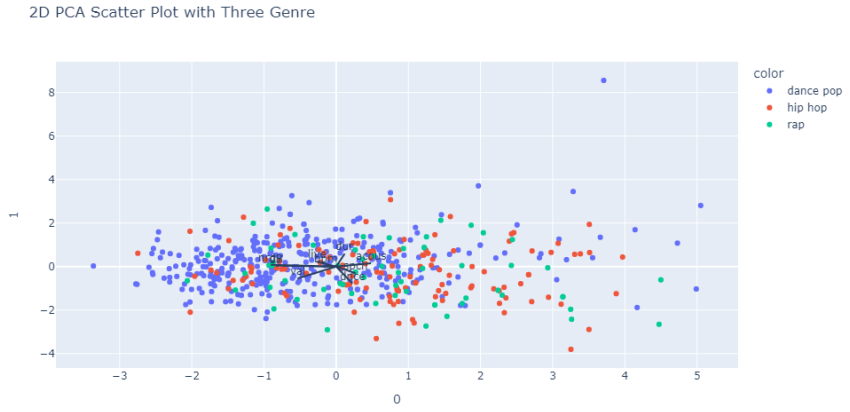

Figure 15: PCA Scatterplot

With two dimensional principal components, a lot of overlapping among genres can be seen indicating classification will be difficult.

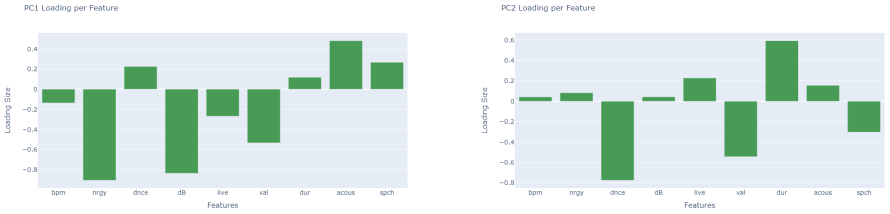

Figure 16: PCA Loadings Per Feature

The magnitude of the numbers within each component indicate how much corresponding feature contributes to each principal component. As seen above, the energy ("nrgy") of the song contributes the most to the first principal component (PC1), followed by loudness ("dB"). This means that energy and loudness of the song are the categories that differentiate a genre of a song the best. Additionally, the measure of how acoustic the song ("acous") has the opposite impact on a genre of a song. We presume this is due to the fact that a song's acoustics has different behavior than energy and loudness of a song, as an acoustic song tends to be less energetic and quiet. Therefore, if a song has high negative PC1 values, then it would most likely to be classified as dance pop song.

5.4 Logistic Regression and XGBoost

We implemented logistic regression and XGBoost machine learning algorithms to predict the genre of the songs using built-in packages. For XGBoost, we set our model with number of estimators set to 100, learning rate set to 0.001, subsample ratio of training set to 0.8, the subsample ratio of columns when constructing each tree set to 0.8, and our evaluation metric was a multiclass negative log-likelihood. For logistic regression, we used the multinomial class.

PCA analysis showed that six components explained 80% of variability. We wanted to observe how effective reducing the dimension of dataset is in classification with 80% of variability. First, we encoded three genres as follows: "dance pop" = 0, "hip hop" = 1, "rap" = 2. We ran the algorithms on two datasets, the original dataset and the standardized and transformed dataset with 6 components from the PCA preprocessing algorithm.

Accuracy for logistic regression on the testing set of the original dataset was 75.4% while the accuracy for the testing set for the PCA set was 71%. Accuracy for XGBoost on the testing set for the original testing set was 73.2% while the accuracy for the testing set for the PCA set was 69.6%. Considering the reduction of the three dimensions, there seems to be no dramatic difference in accuracy between the two. Hence, we could conclude that when we have a larger dimension dataset, PCA would be useful in terms of computing and time as it would reduce the dimension and still give us a similar classification result. It is also improtant to note is that our logistic regression turned out to be slightly more accurate than XGBoost model. Below are confusion matrix comparisons with the left from the original dataset and the right from are PCA dataset.

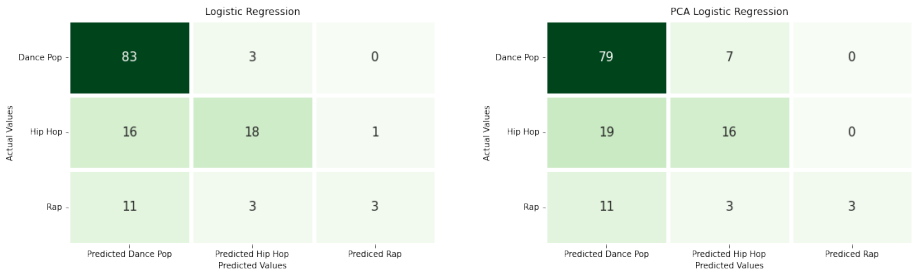

Figure 17: Logistic Regression Confusion Matrix

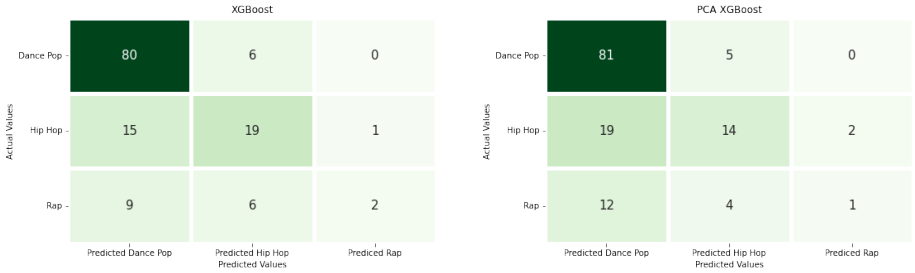

Figure 18: XGBoost Confusion Matrix

6 Conclusion

Our project sought to analyze the top 100 songs on Spotify from 2010-2019. This report summarizes our efforts on two problems: one to create a predicitive model using linear regression and the other a classification problem. For the first, we were able to construct a linear model using LU decomposition and Cholesky decomposition to estimate the regression coefficients and their standard errors. LASSO regression was also used to cut down on the number of predictors and decrease variance. It also had the benefit of decreasing runtime.

Our PCA method was able to reduce our dimension of dataset from 9 to 6, with 80% of variance explained. And our key takeaway was principal component loading for each feature of a song. And we found that energy and loudness of the song are the best categories that differentiate a genre of a song. And based on our knowledge and loadings, we could take this analysis that with the most negative .

We used this preprocessing algorithm from our PCA result to classification algorithm. And we compared our prediction accuracy for both original feature data and PCA standardized and transformed data to see when we use less dimensional data if we can predict genre of a song as well as when we use the original dimension. Our accuracy turned out to be better on the original data, but there was no significant difference in the algorithm on 3 less dimensions. Therefore, PCA would be a great way to optimize our code when we have a larger dimensional dataset than what we have now.

References

[1] “Sklearn.preprocessing.PowerTransformer.” (Scikit) `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PowerTransformer.html`.

[2] “Lasso (Statistics).”(Wikipedia) `https://en.wikipedia.org/wiki/Lasso_(statistics)`

[3] "Multinomial logistic regression" (Wikipedia) `https://en.wikipedia.org/wiki/Multinomial_logistic_regression`

[4] "Gradient descent" (Wikipedia) `https://en.wikipedia.org/wiki/Gradient_descent`

[5] "Gradient boosting" (Wikipedia) https://en.wikipedia.org/wiki/Gradient_boosting

Appendix

For code used in this project, please visit this link to our GitHub repository: https://github.com/anchalamba/STA141C

We optimized our code by:

1. performing parallel computing during a couple sections of our exploratory data analysis (EDA), where for loops were required for (1) creating broader genre categories using regex and (2) creating traces on stacked line graphs
2. utilizing Cholesky decomposition as an improvement when finding estimates for the regression coefficients and also using LASSO regression to decrease the number of predictor variables