

Portfolio

Erik King-Vargas

10/2/23

Table of contents

Index	3
1 Spotify Feature Analysis	4
2 Exploratory Data Analysis	8
3 Data Separation for Modeling	10
4 Finding Separation in Metal and Nonmetal Genres	11
4.1 Metal and Nonmetal	11
4.2 Metal and Country	11
References	14

Index

This is a Quarto book.

This is a line.

To learn more about Quarto books visit <https://quarto.org/docs/books>.

```
1 + 1
```

```
[1] 2
```

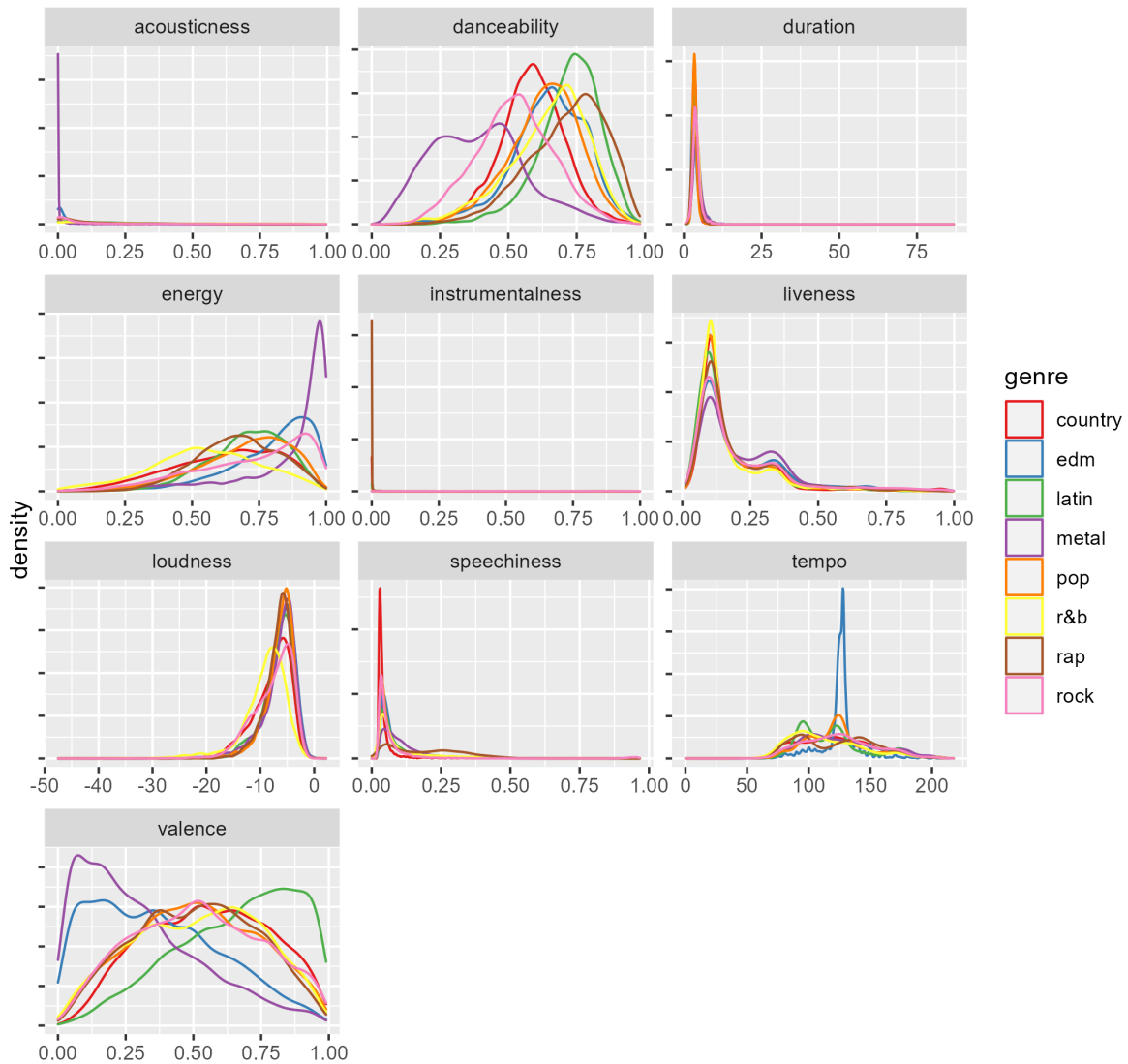
1 Spotify Feature Analysis

One of my favorite hobbies is listening to music. Not just through headphones or in the car, but listening to live music and experiencing all of the energy that comes with live performances. Some time ago I found out that R had a package, `spotifyr` Thomas (2022), that was a wrapper for getting track audio features from Spotify's API. I started exploring the functionality of the package along with a few projects people had done (see)

<https://d2l.ai/index.html>

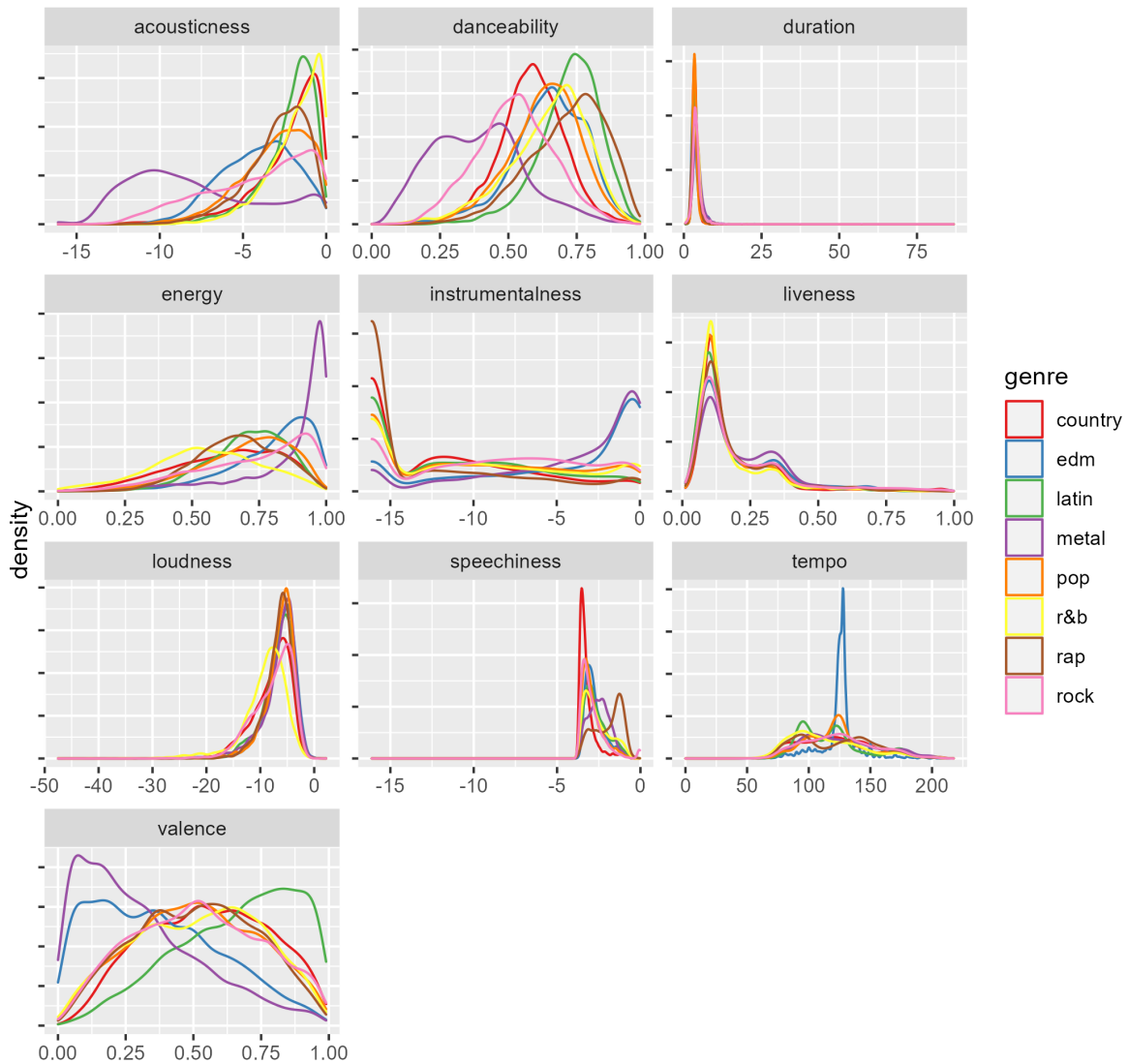
First things first, I needed data before I could analyze. I utilized some function calls to pull down data from Spotify's API with a few predetermined genres of music. From here, I read this in to R to begin exploring what data I had. There were some issues.

Spotify Audio Feature Density - by Genre



Acoustictness seems drastically skewed to the right and concentrated right around 0. Instrumentalness and Loudness seem to also have similar problems. Duration **seems** fine, but there's a large tail that extends to almost an hour and a half. There needs to be some cleaning done here to get the data into an easily analyzed format.

Spotify Audio Feature Density - by Genre



This seems a bit better but duration still looks skewed so lets see why there is such a long tail to the right. First I'll start by looking at how many songs have a duration longer than 10 minutes.

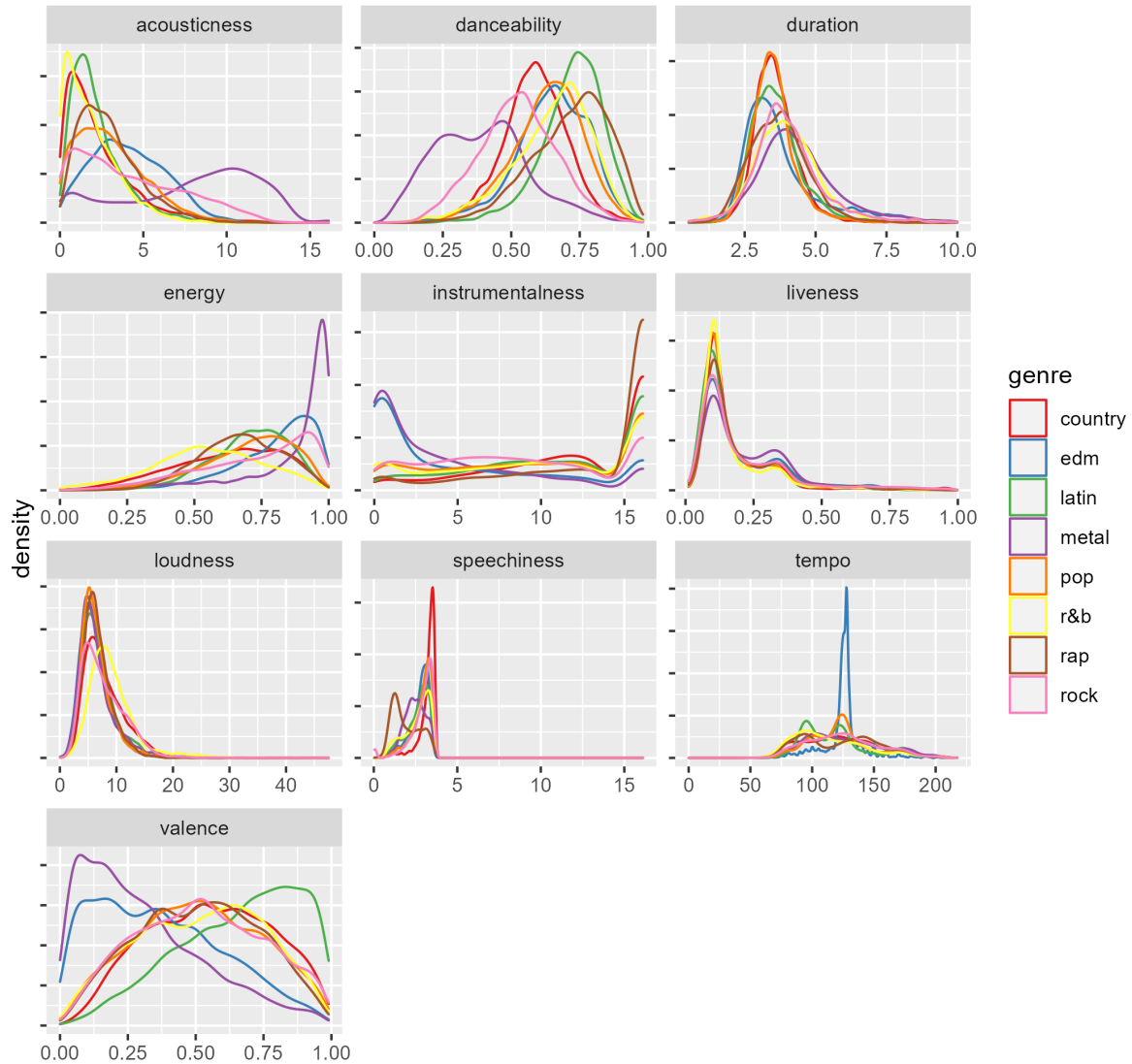
```
Count
1    99
```

There are only 99 songs with a duration longer than 10 minutes and the overall dataset has 36K songs. For the purposes of this analysis it's best to take these out as they don't represent

the majority of the data.

One more look at the density plot before we start taking apart the features and looking towards prediction/classification.

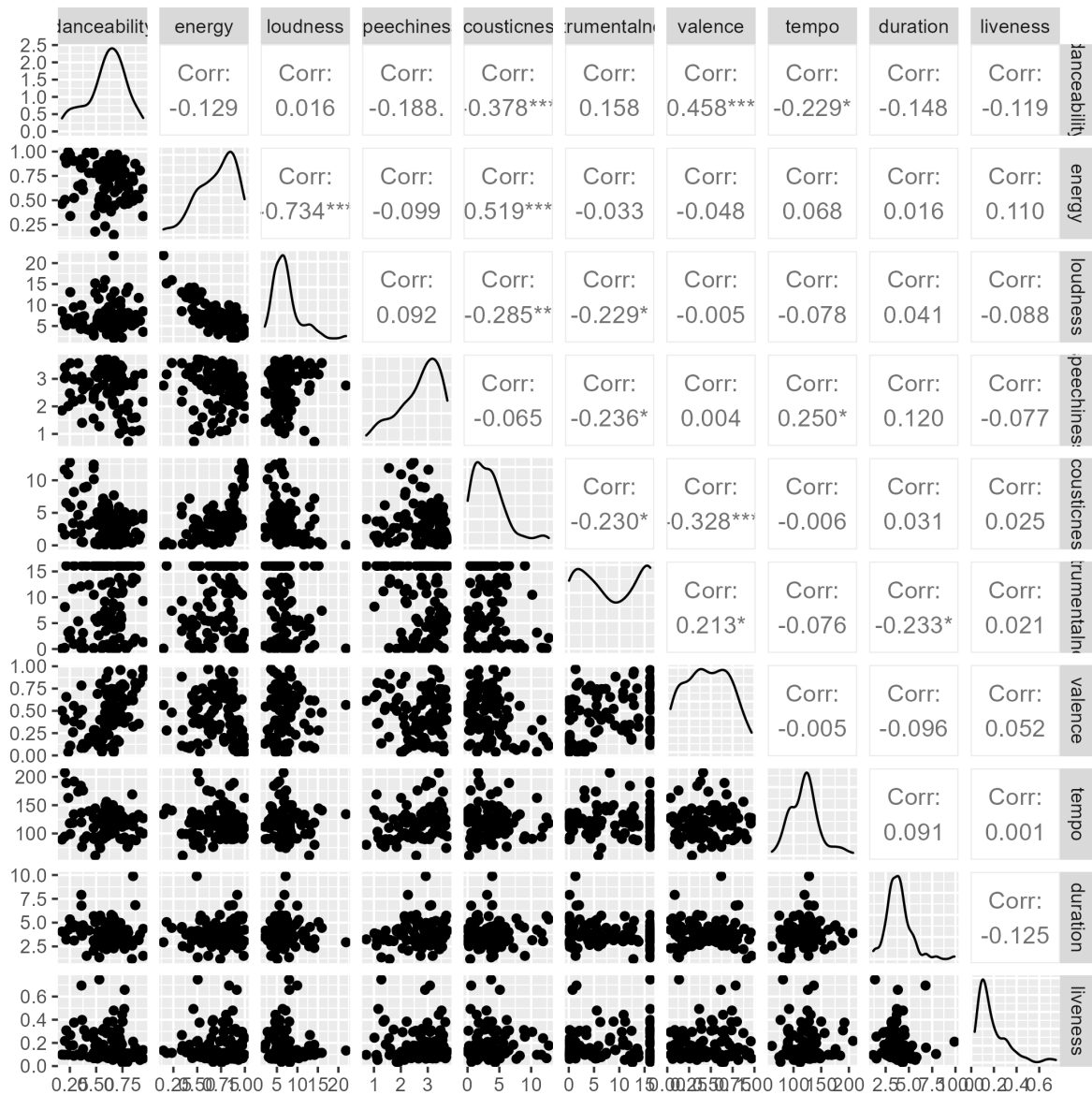
Spotify Audio Feature Density - by Genre



2 Exploratory Data Analysis

Table 2.1: Distribution of Song Counts by Genre

Genre	Count
country	4282
edm	4696
latin	3644
metal	4990
pop	5368
r&b	3924
rap	4751
rock	4265



3 Data Separation for Modeling

4 Finding Separation in Metal and Nonmetal Genres

Performing classification analysis on 8 genres leads to poor accuracy. While each song has many features, the differences may be too small to meaningfully differentiate one genre from another. However, there are still genres that we could certainly recognize as being starkly different. Metal happens to be my favorite genre, but I do listen to country from time to time and it would be hard to mistake one song from each genre for the other. I'll break this experiment out into two parts:

1. One experiment will be looking at metal songs vs. every other genre grouped together as "nonmetal"
2. A second experiment will look at metal vs. country music

4.1 Metal and Nonmetal

4.2 Metal and Country

Starting with logistic regression.

Predicted classes and performance.

Confusion Matrix and Statistics

	Reference	
Prediction	metal	country
metal	1289	142
country	207	1144

Accuracy : 0.8746
95% CI : (0.8617, 0.8866)
No Information Rate : 0.5377
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.7486

McNemar's Test P-Value : 0.0006129

Sensitivity : 0.8616
Specificity : 0.8896
Pos Pred Value : 0.9008
Neg Pred Value : 0.8468
Prevalence : 0.5377
Detection Rate : 0.4633
Detection Prevalence : 0.5144
Balanced Accuracy : 0.8756

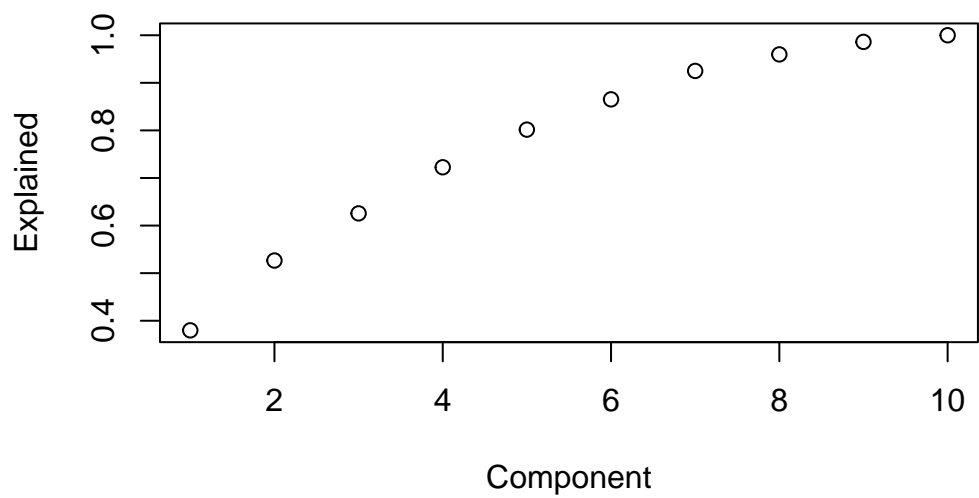
'Positive' Class : metal

Random forest example.

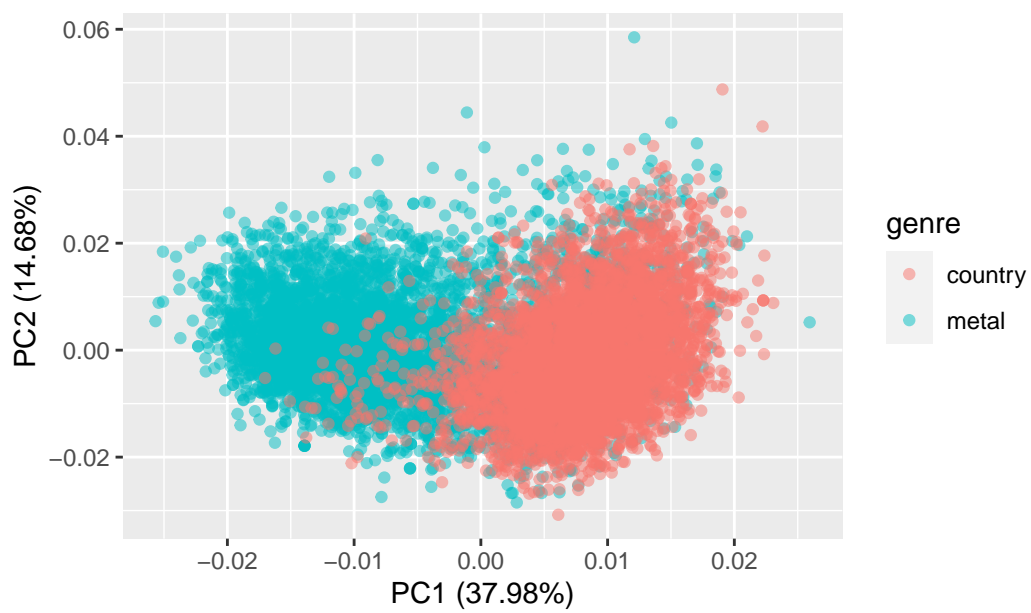
	country	metal
country	1162	124
metal	175	1321

[1] 0.8842559

Cumulative Variance Explained by Components



Plot of Song Data in PC Space



References

Thomas, Charlie. 2022. “Spotifyr Documentation.” 2022. <http://www.rcharlie.com/spotifyr/>.