Challenge II: SPOTIFY DATASET

Khanh Tra Nguyen Tran

11/02/2023

I. INTRODUCTION

Nowadays, Spotify becomes one of the largest music streaming platforms in the world, emerging as a cultural phenomenon and captivating music enthusiasts worldwide with its vast library of songs and artists as well as user-friendly and attractive interface. In this era, where music consumption has become an integral part of daily life, Spotify are considered by many smart device users as an indispensable app that they utilize to access to an extensive collection of tracks spanning various genres and artists. While the platform excels in delivering personalized playlists and recommendations, an intriguing question arises for those users seeking to unravel the dynamics of music popularity. Some common questions are Is predicting a song's popularity solely contingent on the reputation of its artist or genre, or can we delve deeper into the intricacies of the song itself, such as the "meaningfulness" or "brightness" of the lyrics or the hidden formula for a viral song lays on the instrument? Acknowledging this, Spotify provides their analysis with the attributes of each song they have on the platform so the curious users can retrieve these precious data and conduct the research on their own.

In this challenge, utilizing this powerful datasert, I will delve into the questions: Whether we can use the attributes of the songs to predict the genre of a track without considering any other genre-related attributes (artists who are famous with a specific genre, the main genre of the album this track belongs to,...).

II. APPROACH

1. Dataset:

Wrangling the data:

In the raw dataset, we have more than 30,417 observational units and 23 variables. For the `track_album_release_date`, it initially has the type of , which is not really suitable to use with the date type, so I converted this type into using ymd(). When I conducted this step, `track_album_release_date` has some rows with missing values. Therefore, I dropped all the row with missing values. Also, I wanted to specify the mode of each song instead of the number of 1s (major scale) and 0s (minor scale). Since the dataset is so big, I decided to only use the tracks released after 2015 to narrow down the dataset to about 18,000 rows and removed some redundant variables that may make the model work less efficiently. Finally, I converted the categorical variables (playlist_genre, mode) into type.

Important Variable Description

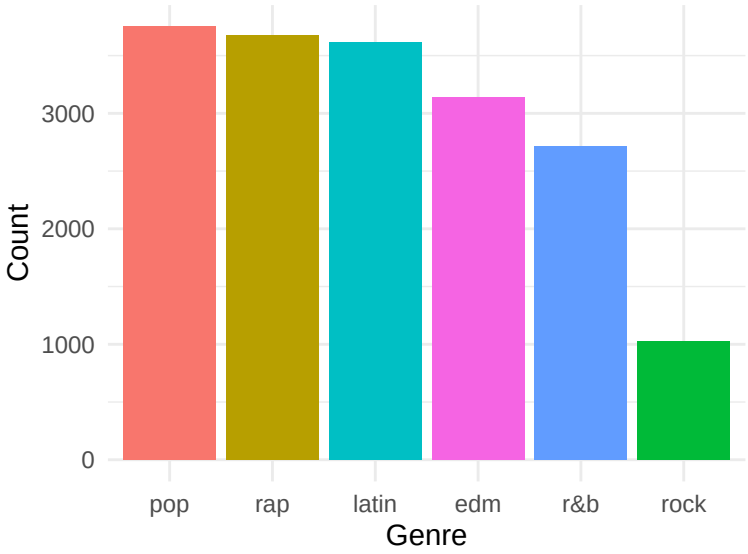
In the modified dataset *spotify_cleaned*, these are variables under consideration: `acousticness`, `danceability`, `playlist_genre`, `valence`, `tempo` , `speechiness`, `mode`, `loudness`, `instrumentalness`, `energy`, `duration_ms`, and `track_popularity`.

Variable Description Reference

The genres available in this dataset:

```
##
##   pop   rap  rock latin   r&b   edm
##  3750  3674  1025  3616  2711  3138
```

Distribution of genres:



Pop is the genre which has the largest number of tracks while rap is the least one. The difference between these genres is not really significant for us to consider to lump the minority groups.

2. Training and testing data sets:

I set the seed to make sure that everytime I rerun this chunk of code, it will produce the same training and testing datasets. I chose to get 80% observations as training units and 20% observations as testing units since I assumed the more observations utilized for training the models, the more accurate the model is.

III. CLASSIFICATION MODELING

I attempted to base on other variables to predict the genre of a track. I decided to use lasso classification, ridge classfication, random forest, and boosting.

Since the recipe will remain the same for every modelling method, I have the recipe for this classification mission:

Recipe

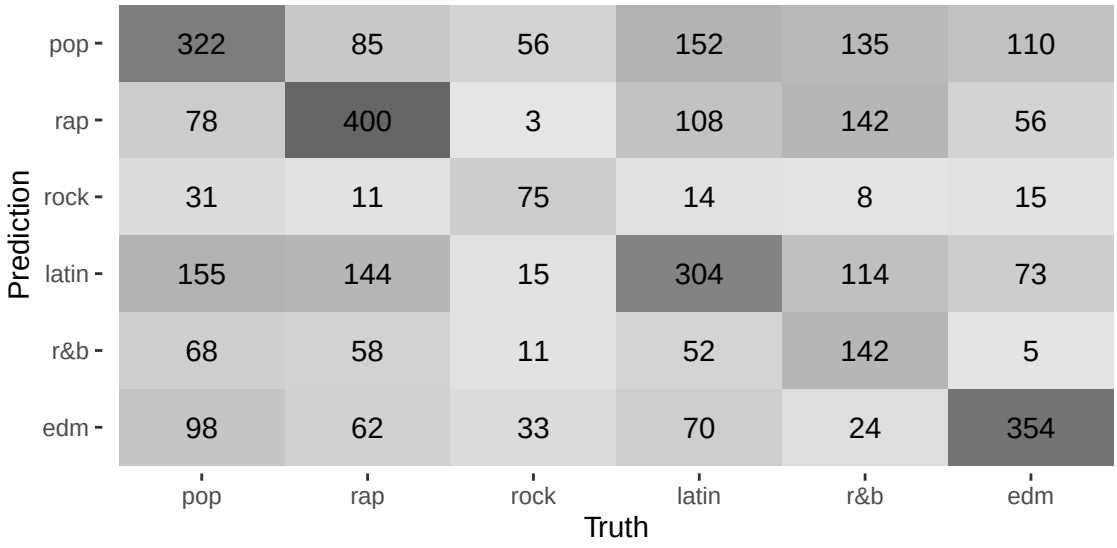
For the cross-validation to optimize the parameters, I used 20-fold cross validation and penalty grid range from -10 to 10:

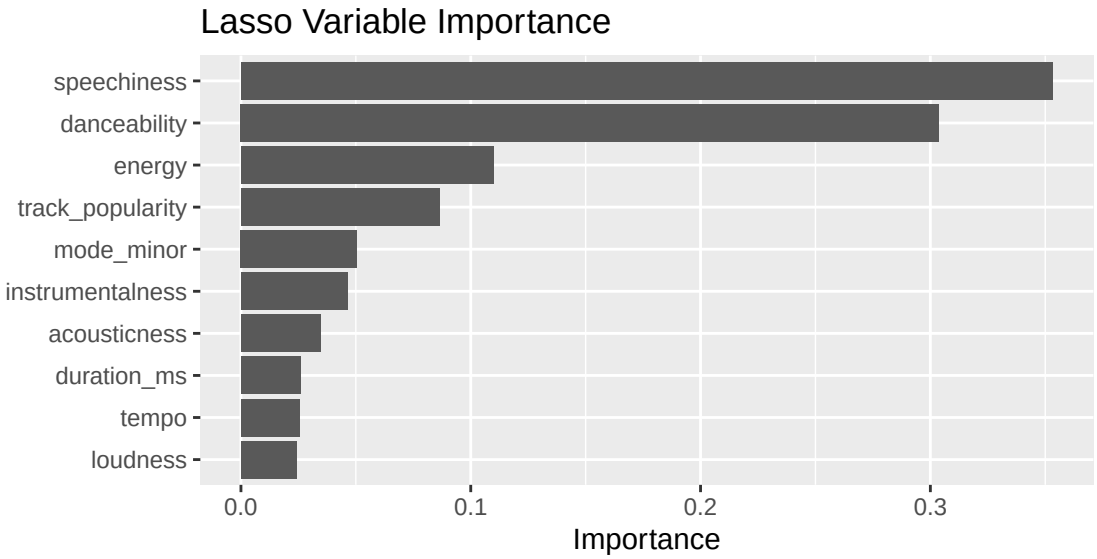
Lasso classification:

```
## # A tibble: 1 x 2
##       penalty .config
```

```
##          <dbl> <chr>
## 1 0.0000000001 Preprocessor1_Model01
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy multiclass     0.446
```



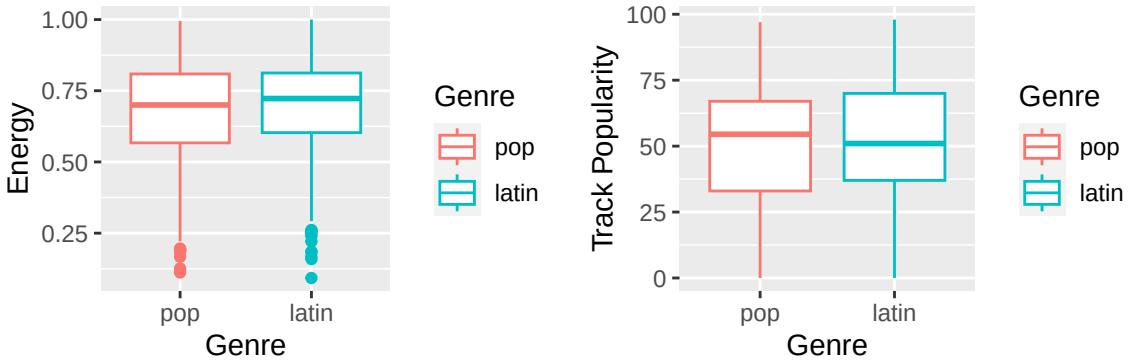


With this Lasso classification mode, the top five most important coefficients are `speechiness`, `danceability`, `energy`, `track_popularity`, and `duration_ms`.

We have the accuracy of this model: 0.447. Therefore, it predicts correctly the genre for 44.6% or about 1,598 out of 3583 observations in the testing data set.

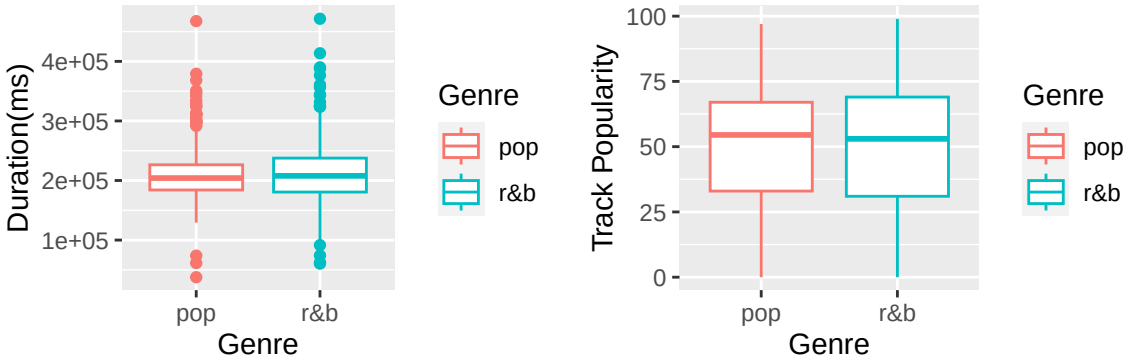
The two genres got misclassified the most:

a. many **pop** tracks got misclassified as **latin** tracks:



The two genres have the similar ranges as well as distribution of `energy` and `track_popularity` (2 of 5 most important variables).

b. many **r&b** track got misclassified as **pop** tracks:



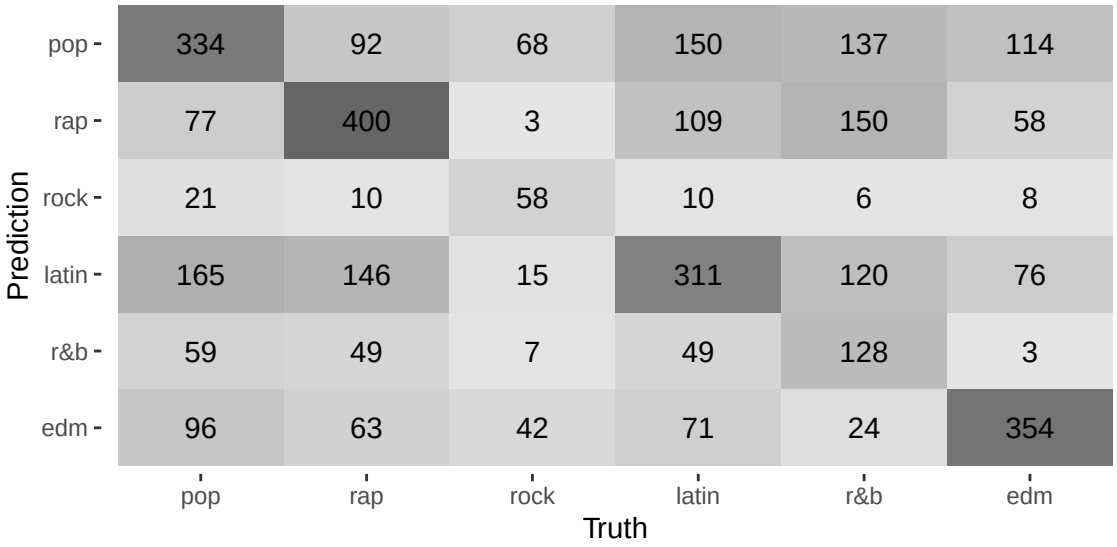
The two genres have the similar ranges as well as distribution of `duration_ms` and `track_popularity` (2 of 5 most important variables).

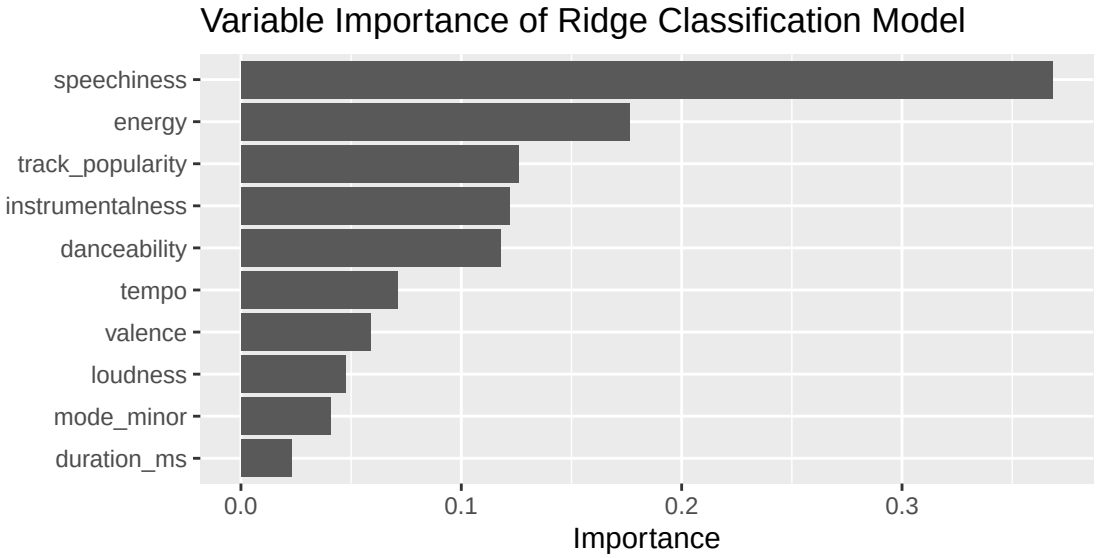
Ridge classification:

```
## # A tibble: 5 x 7
##        penalty .metric  .estimator  mean     n std_err .config
##          <dbl> <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 0.0000000001 accuracy multiclass 0.445    20 0.00443 Preprocessor1_Model01
## 2 0.0000000167 accuracy multiclass 0.445    20 0.00443 Preprocessor1_Model02
## 3 0.00000278   accuracy multiclass 0.445    20 0.00443 Preprocessor1_Model03
## 4 0.000464     accuracy multiclass 0.445    20 0.00443 Preprocessor1_Model04
## 5 0.0774       accuracy multiclass 0.431    20 0.00498 Preprocessor1_Model05

## # A tibble: 1 x 2
##        penalty .config
##          <dbl> <chr>
## 1 0.0000000001 Preprocessor1_Model01

## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy multiclass     0.442
```





With this Ridge classification mode, the top five most important coefficients are `speechiness`, `energy`, `track_popularity`, `instrumentalness`, and `danceability`.

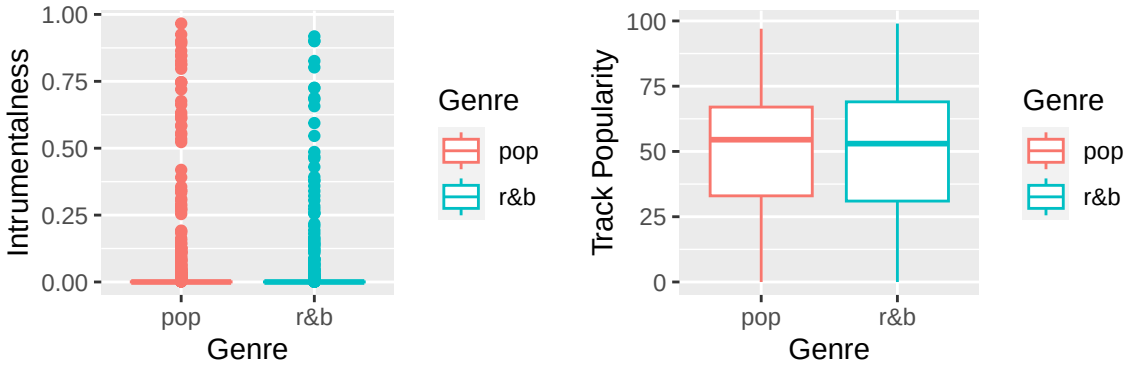
We have the accuracy of this model: 0.442. Therefore, it predicts correctly 44.2% or 1583 out of 3583 observations in the testing data set.

The two genres got misclassified the most:

a. many **pop** tracks got misclassified as **latin** tracks

According to the plots a in Lasso Classification, these two genres have similar distribution of `energy` and `track_popularity`.

b. many **r&b** track got misclassified as **pop** tracks



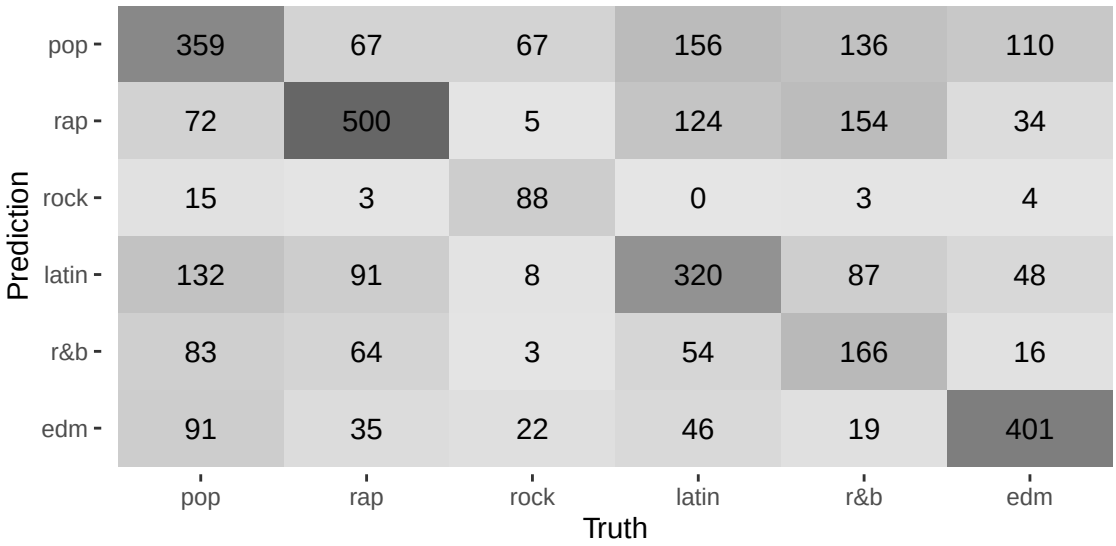
The two genres have the similar ranges as well as distribution of `instrumentalness` and `track_popularity` (2 of 5 most important variables).

Random Forest:

Since this training data set has a really large amount of observational units, I decided to lower down the number of trees for this model down to 100 and only use 10-fold cross-validation. I tuned the min_n and mtry parameters to get the most efficient model.

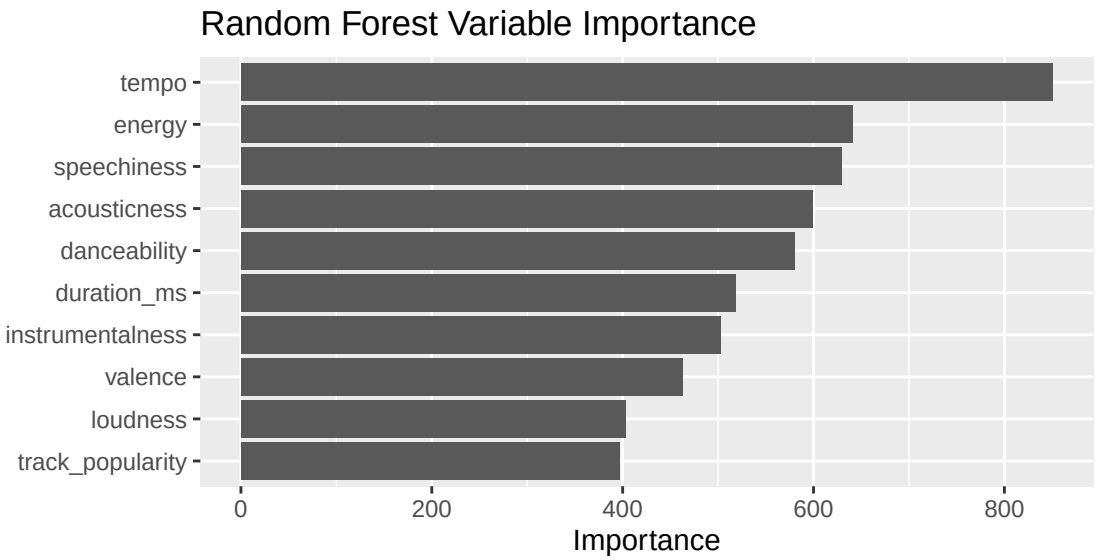
```
## # A tibble: 1 x 3
##    mtry min_n .config
##   <int> <int> <chr>
## 1     2    27 Preprocessor1_Model03
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy multiclass     0.512
```



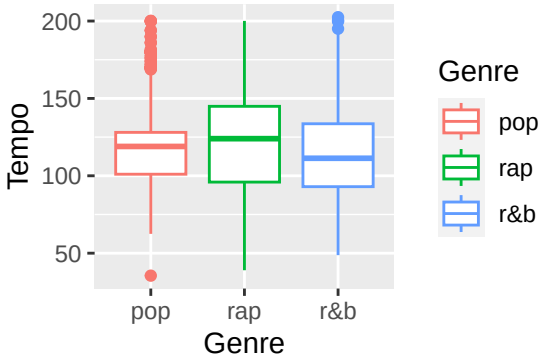
We have the accuracy of this model: 0.515. Therefore, it predicts correctly 51.5% or 1680 out of 3304 observations in the testing data set.

The variables that are most important in the dataset: `tempo`, `energy`, `speechiness`, `acousticness`, `danceability`.



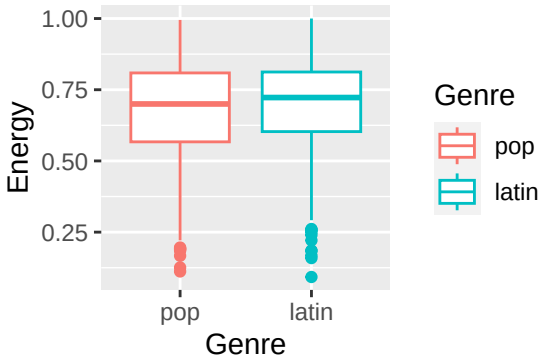
The two genres got misclassified the most:

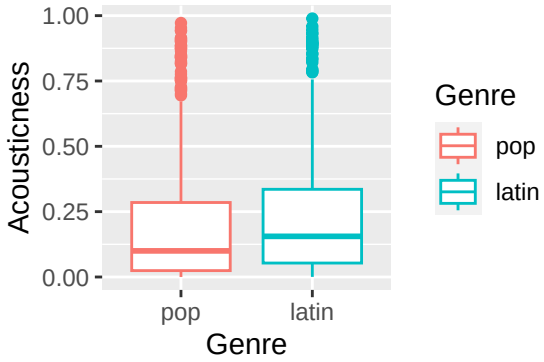
a. Many **r&b** tracks got misclassified as **rap** and **pop** tracks.



These three have kind of the same range and distribution for `tempo` (the most important variable).

b. Many **latin** tracks got misclassified as **pop** tracks.



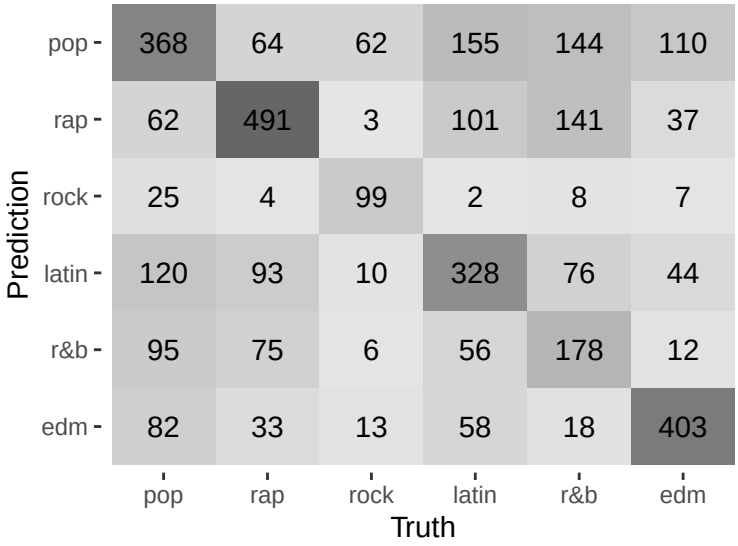


Two genres have the same range of `energy` and `acousticness`. (2 of 5 most important variables)

Boosting:

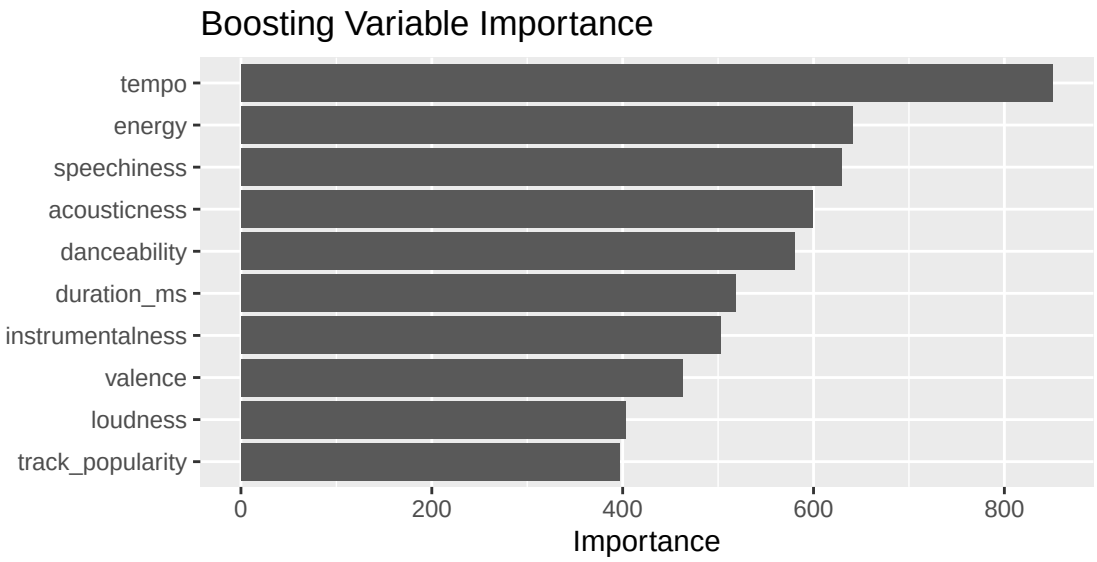
Since this training dataset has a large number of observations, I only used 10-fold cross validation and tuned tree numbers (ranging from 50 to 100 trees) and learning rate (from -3 to 0) of the model. I specify levels as 5.

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy multiclass     0.521
```



We have the accuracy of this model: 0.521. Therefore, it predicts correctly 52.1% or 1866 out of 3583 observations in the testing data set.

The variables that are most important in the dataset: `tempo`, `energy`, `speechiness`, `acousticness`, and `danceability`.



The two genres got misclassified the most:

a. Many **r&b** tracks got misclassified as **rap** and **pop** tracks.

According to the plot a in Random Forest, the range of variable `tempo` of these three genres are similar to others.

b. Many **latin** tracks got misclassified as **pop** tracks.

According to the plots b in Random Forest, the range of two variables `acousticness` and `energy` of two genres are similar to others.

IV. CONCLUSION

Among four classification models to classify the genre of the track, the Boosting works most efficiently, followed by the Lasso Regression and Ridge Regression.

Table 1: Classification Result

Models	Fold	Accuracy	Variables
Boosting	10	0.527	tempo, speechiness, energy, acousticness, danceability
Random Forest	10	0.515	tempo, speechiness, danceability, energy, acousticness
Lasso	20	0.447	speechiness, energy, track_popularity, danceability, instrumentalness
Regression	20	0.442	speechiness,danceability,energy,track_popularity,duration_ms

In conclusion, the observed limitations in achieving a high level of accuracy in the classification model can be attributed to several factors. Firstly, a notable challenge arises from the absence of discernible trends or specific patterns within the variables for each genre across the majority of predictors. While certain variables may be identified as more significant, their specificity within each genre remains insufficient, leading to an elevated risk of misclassification. Secondly, the classification task involves six distinct classes, further complicating the predictive task.

Furthermore, the expansive size of the initial dataset, comprising approximately 19,000 observational units, introduces additional complexities. The voluminous nature of the data raises concerns about data quality, as the presence of outliers, missing values, noise, or errors may exert an unexpected influence on the models' accuracy. As a consequence, addressing these issues becomes imperative to enhance the robustness and reliability of the classification model.

Though the accuracy is not high (>80%), these models still can predict precisely a large amount of songs' genres in the testing dataset (about 1,500 observational units). For improvement, I think of processing the dataset again to remove the outliers in each variable, and employing robust cross-validation techniques to assess the model's performance across different subsets of the data.