Mood-Based Song Classifier. *

Álvaro Novillo Correas *Universidad Carlos III*

In this article, following a similar procedure to (Trochidis et al., 2008), we apply several machine learning techniques to classify songs into four different moods: energetic, happy, calm and sad. We then implement a Shiny app that uses Python to scrape data from Spotify, given an artist name or playlist URL, and perform an analysis of the different characteristics and emotions that the artist's discography or playlist has.

Keywords: pandoc, r markdown, knitr

Introduction.

Music, among other art forms, has an special ability to evoke emotions, shaping our experiences and perceptions. In this article, we will delve into the intricate relationship between music and emotions, using machine learning techniques to analyze and categorize songs based on their emotional content.

Building on previous research (Trochidis et al., 2008), we will classify songs into four primary emotional categories: energetic, happy, calm, and sad. To achieve this, we'll leverage a dataset[1] of 686 previously classified songs and develop a robust model capable of discerning the encoded emotionality within musical compositions.

Table 1: Pre-classified dataset used to train the model

name	artist	album	id	danceability	acousticness	energy	instrumentalness	liveness	valence	loudness	speechiness	tempo	mood
1999	Prince	1999	2H7PHVdQ3mXqEHXcvclTB0	0.866	0.13700	0.7300	0.00e+00	0.0843	0.625	-8.201	0.0767	118.523	Happy
23	Blonde Redhead	23	4HIwL9ii9CcXpTOTzMq0MP	0.381	0.01890	0.8320	1.96e-01	0.1530	0.166	-5.069	0.0492	120.255	Sad
9 Crimes	Damien Rice	9	5GZEeowhvSieFDiR8fQ2im	0.346	0.91300	0.1390	7.73e-05	0.0934	0.116	-15.326	0.0321	136.168	Sad
99 Luftballons	Nena	99 Luftballons	6HA97v4wEGQ5TUClRM0XLc	0.466	0.08900	0.4380	5.60e-06	0.1130	0.587	-12.858	0.0608	193.100	Happy
A Boy Brushed Red Living In Black And White	Underoath	They're Only Chasing Safety	47IWLfIKOKhFnz1FUEUIkE	0.419	0.00171	0.9320	0.00e+00	0.1370	0.445	-3.604	0.1060	169.881	Energetic
A Burden to Bear	Emmanuelle Rimbaud	A Burden to Bear	67DOFCrkcQaLp5yhzF8Y8N	0.394	0.99500	0.0475	9.55e-01	0.1050	0.172	-26.432	0.0720	71.241	Calm

We will follow a systematic approach, using different machine learning algorithms to train a model that can accurately identify the mood of a song. We achieve this by utilizing key parameters such as danceability, acousticness, energy, instrumentalness, liveness, valence, loudness and speechiness, obtained directly from the Spotify API (*Spotify Developer Documentation*, N.d.)

After obtaining the best model possible, we will apply it in a basic Shiny app, where we've built a Python-based scraping algorithm to gather essential features and attributes of songs using Spotify API (*Spotify Developer Documentation*, N.d.), allowing us to predict the corresponfing mood of each song, and analyze the data desired by the user.

Exploratory Data Analysis

After an initial inspection of the dataset, we will take first all the variables obtained from the Spotify API to fit our model. Fig. 1 contains a pairs plot of each feature with respect to each mood. Some of the features, such as danceability, or tempo, appears to have a gaussian distribution for all

*Replication files are available on the author's Github account (https://github.com/AlvaroNovillo/music_mood).
Current version: febrero 20, 2024; **Corresponding author**: novillocorreasalvaro@gmail.com.

[1]The dataset used to train our model can be found in Kaggle

modes (centered in different values). Others, such as valence or loudness, have different skewed distributions.

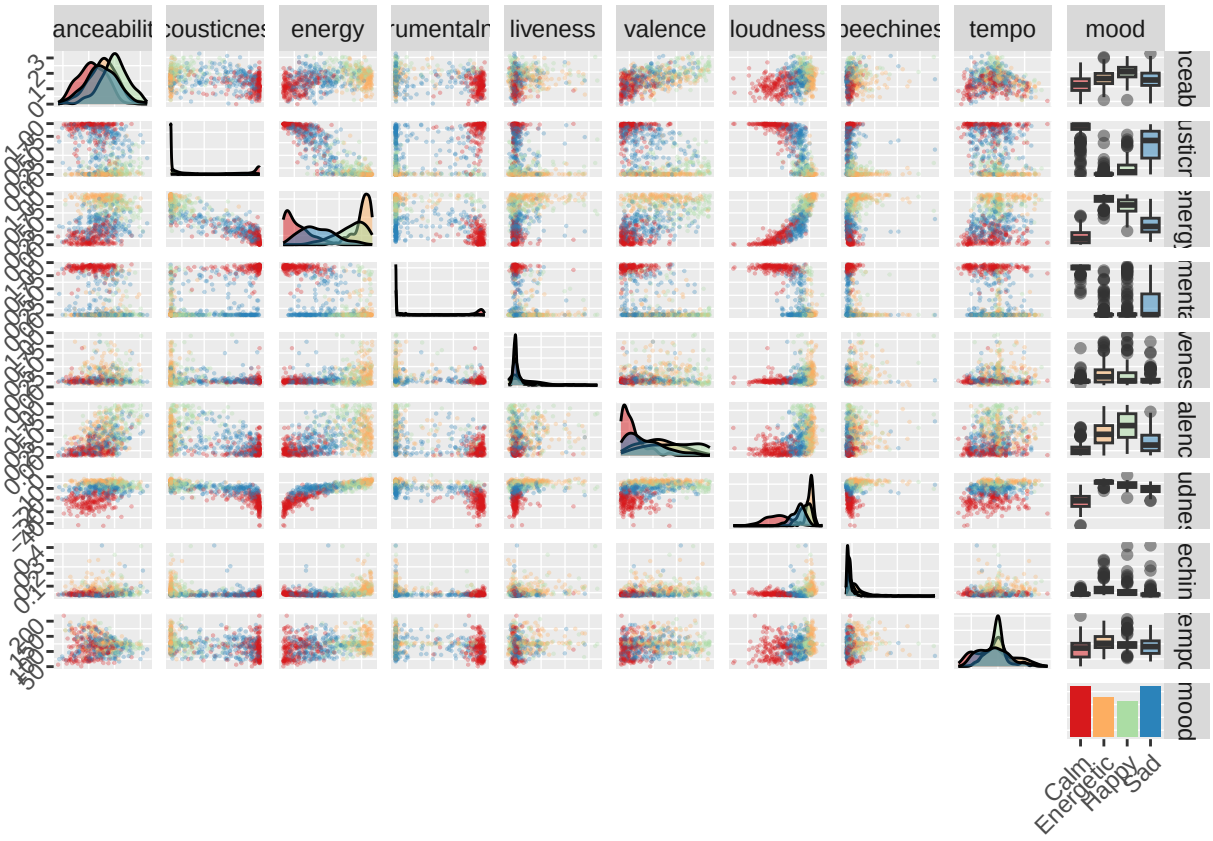

Figure 1: Features inspection with respect to each mood labeled.

Also, from the scatter plots appearing in 1, we can see that some of the variables are highly correlated. 2 contains the correlation matrix for the variables, presenting both negative and positive correlation among some of the features, such as loudness with energy (positive) and aciusticness with energy (negative).

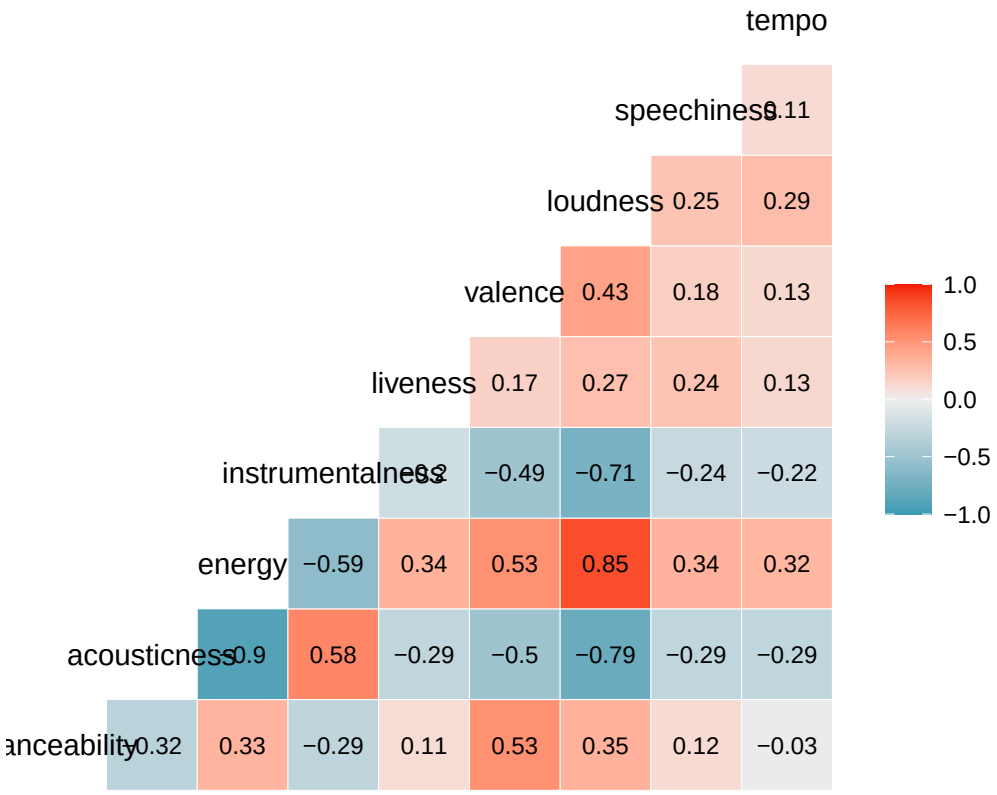

Figure 2: Correlation plot of the features.

If we inspect closer the distribution of moods in our dataset, (Fig.3), we can see that our dataset is pretty balanced, containing more than enough songs to characterize each mood.

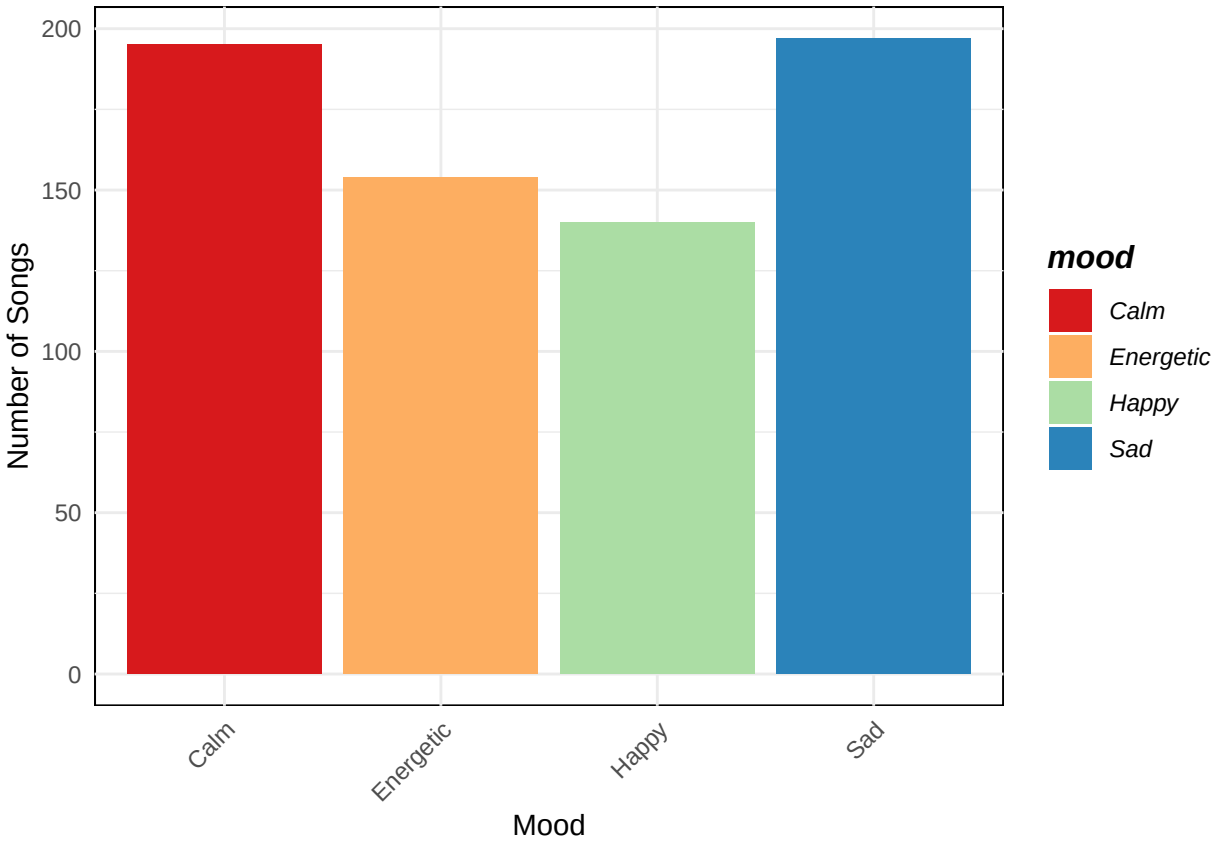

Figure 3: Histogram of the amount of songs classified in each emotion

Lastly, Fig 4 shows a characterization of each of the moods using a Radar plot. As you can see, calm songs are characterized by being both heavily acoustic and instrumental, while happy songs are highly danceable and energetic, being moderatelly loud. Sad songs seems to be very valanced in every aspect except from speechiness and energetic songs are characterized by being energetic and loud.

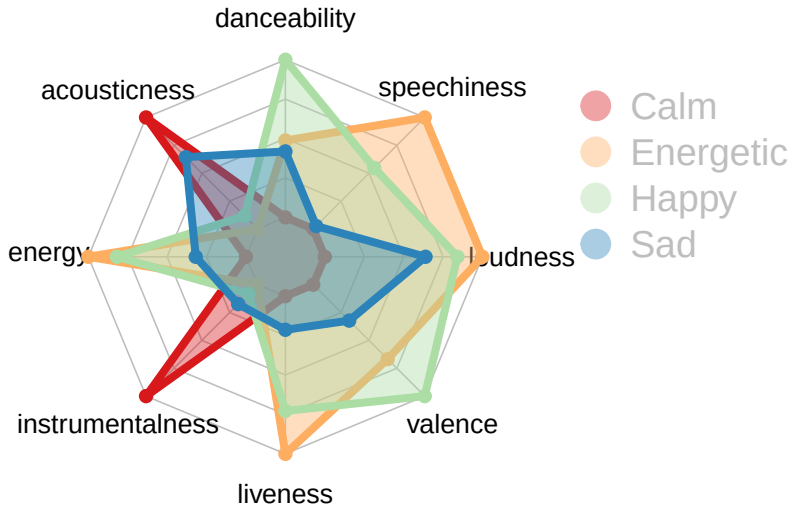

Figure 4: Radar plot showcasing the average values of the features that characterize each of the moods

Methodolody

To train our models, we have split our dataset into a training set (80%) and a test set (20%). Shuffling is not necessary as the dataset is not ordered. Our approach to finding the best model involves fitting several models using all numerical features available in our dataset, including 'danceability', 'acousticness', 'energy', 'instrumentalness', 'liveness', 'valence', 'loudness', 'speechiness', and 'tempo'. Afterwards, we will select the best performing model and improve it through feature selection and hyperparameter tuning. The model with the highest performance will then be refitted with the entire dataset and integrated into our Shiny app. To fit all the models, `caret` package (Kuhn and Max, 2008) will be used, 5-folds CV will be applied, and we will center and scale the predictors.

Multinomial Logistic Regression

We will begin by applying the most basic model, which is a logistic regression. In this case, we will use a multinomial logistic regression since our categories are not binary.

```
## Penalized Multinomial Regression
##
```

```
## 550 samples
##   9 predictor
##   4 classes: 'Calm', 'Energetic', 'Happy', 'Sad'
##
## Pre-processing: centered (9), scaled (9)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 440, 440, 440, 439, 441
## Resampling results across tuning parameters:
##
##   decay  Accuracy   Kappa
##   0e+00  0.7982580  0.7286732
##   1e-04  0.7982580  0.7286732
##   1e-01  0.7981922  0.7287540
##
## Kappa was used to select the optimal model using the largest value.
## The final value used for the model was decay = 0.1.
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Calm Energetic Happy Sad
##   Calm        37         0     0   2
##   Energetic    0        23     8   0
##   Happy        0         7    17   2
##   Sad          2         0     3  35
##
## Overall Statistics
##
##                Accuracy : 0.8235
##                  95% CI : (0.7489, 0.8835)
##     No Information Rate : 0.2868
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7627
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Calm Class: Energetic Class: Happy Class: Sad
## Sensitivity               0.9487           0.7667       0.6071     0.8974
## Specificity               0.9794           0.9245       0.9167     0.9485
## Pos Pred Value            0.9487           0.7419       0.6538     0.8750
## Neg Pred Value            0.9794           0.9333       0.9000     0.9583
## Prevalence                0.2868           0.2206       0.2059     0.2868
## Detection Rate            0.2721           0.1691       0.1250     0.2574
## Detection Prevalence      0.2868           0.2279       0.1912     0.2941
## Balanced Accuracy         0.9640           0.8456       0.7619     0.9229
```

Although multinomial logistic regression model appears to be effective in classifying songs into their respective moods, with an accuracy of ~80% and $\kappa = 0.76$,we will keep trying more appropriate methods for non-binary classification problems, such as LDA for $p > 1$. Before that, lets have a look at the coefficients computed for each variable

Table 2: Coefficients for mood = Energetic

	(Intercept)	danceability	acousticness	energy	instrumentalness	liveness	valence	loudness	speechiness	tempo
Coefficient	-1.8037649	0.5121807	-1.9035712	4.1097341	-4.4191853	0.6185421	1.1889409	3.4788527	-0.0894802	0.4523645
Std. Errors	1.0888229	0.4516770	0.8134856	1.1198090	0.6506191	0.6403064	0.5921362	1.1964151	0.6271437	0.4224320
z stat	-1.6566192	1.1339536	-2.3400183	3.6700314	-6.7922767	0.9660095	2.0078844	2.9077304	-0.1426789	1.0708577
p value	0.0975965	0.2568140	0.0192828	0.0002425	0.0000000	0.3340394	0.0446556	0.0036406	0.8865438	0.2842334

Table 3: Coefficients for mood = Happy

	(Intercept)	danceability	acousticness	energy	instrumentalness	liveness	valence	loudness	speechiness	tempo
Coefficient	1.7063251	1.0854503	-1.3099355	3.123178	-3.4162837	0.5534468	1.6673193	0.9929464	-0.1377737	0.5614269
Std. Errors	0.7922969	0.4175990	0.5734972	1.015434	0.5739915	0.6304147	0.5733037	1.0095547	0.6175027	0.4023728
z stat	2.1536436	2.5992650	-2.2841182	3.075708	-5.9518015	0.8779090	2.9082652	0.9835489	-0.2231144	1.3952904
p value	0.0312681	0.0093424	0.0223646	0.002100	0.0000000	0.3799931	0.0036344	0.3253374	0.8234465	0.1629283

Table 4: Coefficients for mood = Sad

	(Intercept)	danceability	acousticness	energy	instrumentalness	liveness	valence	loudness	speechiness	tempo
Coefficient	3.8190113	-0.1399674	-0.2790502	0.2507406	-2.7361226	0.2842616	1.2273135	2.0233308	-0.3899211	-0.1526533
Std. Errors	0.7077692	0.3288764	0.4355993	0.8415041	0.4808576	0.5943572	0.5285823	0.6882719	0.6016057	0.2987553
z stat	5.3958429	-0.4255928	-0.6406121	0.2979672	-5.6900892	0.4782673	2.3218967	2.9397261	-0.6481340	-0.5109644
p value	0.0000001	0.6704046	0.5217748	0.7657282	0.0000000	0.6324600	0.0202385	0.0032850	0.5168983	0.6093760

As we can see in the tables above, some of the coefficients do not affect the dependent variable. This is due to the multicolinearity found before in Fig. 2.

Linear Discriminant Analysis (LDA, p > 1)

```
## Linear Discriminant Analysis
##
## 550 samples
##   9 predictor
##   4 classes: 'Calm', 'Energetic', 'Happy', 'Sad'
##
## Pre-processing: centered (9), scaled (9)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 440, 440, 440, 439, 441
## Resampling results:
##
##   Accuracy   Kappa
##   0.7745379  0.6975865

## Confusion Matrix and Statistics
##
```

```
##            Reference
## Prediction  Calm Energetic Happy Sad
##   Calm        36         0     0   2
##   Energetic    0        23    10   0
##   Happy        0         7    16   5
##   Sad          3         0     2  32
##
## Overall Statistics
##
##                Accuracy : 0.7868
##                  95% CI : (0.7083, 0.8523)
##     No Information Rate : 0.2868
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7141
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Calm Class: Energetic Class: Happy Class: Sad
## Sensitivity               0.9231           0.7667       0.5714     0.8205
## Specificity               0.9794           0.9057       0.8889     0.9485
## Pos Pred Value            0.9474           0.6970       0.5714     0.8649
## Neg Pred Value            0.9694           0.9320       0.8889     0.9293
## Prevalence                0.2868           0.2206       0.2059     0.2868
## Detection Rate            0.2647           0.1691       0.1176     0.2353
## Detection Prevalence      0.2794           0.2426       0.2059     0.2721
## Balanced Accuracy         0.9512           0.8362       0.7302     0.8845
```

the LDA and logistic regression predictions are almost identical, as Section 4.5 of (James et al., 2021) discusses.

Quadratic Discriminant Analysis (QDA)

```
## Quadratic Discriminant Analysis
##
## 550 samples
##   9 predictor
##   4 classes: 'Calm', 'Energetic', 'Happy', 'Sad'
##
## Pre-processing: centered (9), scaled (9)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 440, 440, 440, 439, 441
## Resampling results:
##
##   Accuracy   Kappa
##   0.7835961  0.7091766
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Calm Energetic Happy Sad
##   Calm        37         0     0   2
##   Energetic    0        20     9   0
##   Happy        0         9    16   4
##   Sad          2         1     3  33
##
## Overall Statistics
##
##                Accuracy : 0.7794
##                  95% CI : (0.7003, 0.8459)
##     No Information Rate : 0.2868
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7037
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Calm Class: Energetic Class: Happy Class: Sad
## Sensitivity               0.9487           0.6667       0.5714     0.8462
## Specificity               0.9794           0.9151       0.8796     0.9381
## Pos Pred Value            0.9487           0.6897       0.5517     0.8462
## Neg Pred Value            0.9794           0.9065       0.8879     0.9381
## Prevalence                0.2868           0.2206       0.2059     0.2868
## Detection Rate            0.2721           0.1471       0.1176     0.2426
## Detection Prevalence      0.2868           0.2132       0.2132     0.2868
## Balanced Accuracy         0.9640           0.7909       0.7255     0.8921
```

Up to this point, the three models fitted seems to perform very similarly on our dataset. To wrap up the model testing, we will fit try Naive Bayes

Naive Bayes

```
## Naive Bayes
##
## 550 samples
##   9 predictor
##   4 classes: 'Calm', 'Energetic', 'Happy', 'Sad'
##
## Pre-processing: centered (9), scaled (9)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 440, 440, 440, 439, 441
## Resampling results across tuning parameters:
##
```

```
##   usekernel  Accuracy   Kappa
##   FALSE      0.7964065  0.7267479
##    TRUE      0.7818110  0.7076660
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Kappa was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = FALSE and adjust
##  = 1.
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Calm Energetic Happy Sad
##   Calm        35         0     0   2
##   Energetic    0        23    10   0
##   Happy        0         6    15   6
##   Sad          4         1     3  31
##
## Overall Statistics
##
##                Accuracy : 0.7647
##                  95% CI : (0.6844, 0.8332)
##     No Information Rate : 0.2868
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6843
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Calm Class: Energetic Class: Happy Class: Sad
## Sensitivity               0.8974           0.7667       0.5357     0.7949
## Specificity               0.9794           0.9057       0.8889     0.9175
## Pos Pred Value            0.9459           0.6970       0.5556     0.7949
## Neg Pred Value            0.9596           0.9320       0.8807     0.9175
## Prevalence                0.2868           0.2206       0.2059     0.2868
## Detection Rate            0.2574           0.1691       0.1103     0.2279
## Detection Prevalence      0.2721           0.2426       0.1985     0.2868
## Balanced Accuracy         0.9384           0.8362       0.7123     0.8562
```

We can compare the four models applyied using `resamples` function from the `caret` package (Kuhn and Max, 2008)

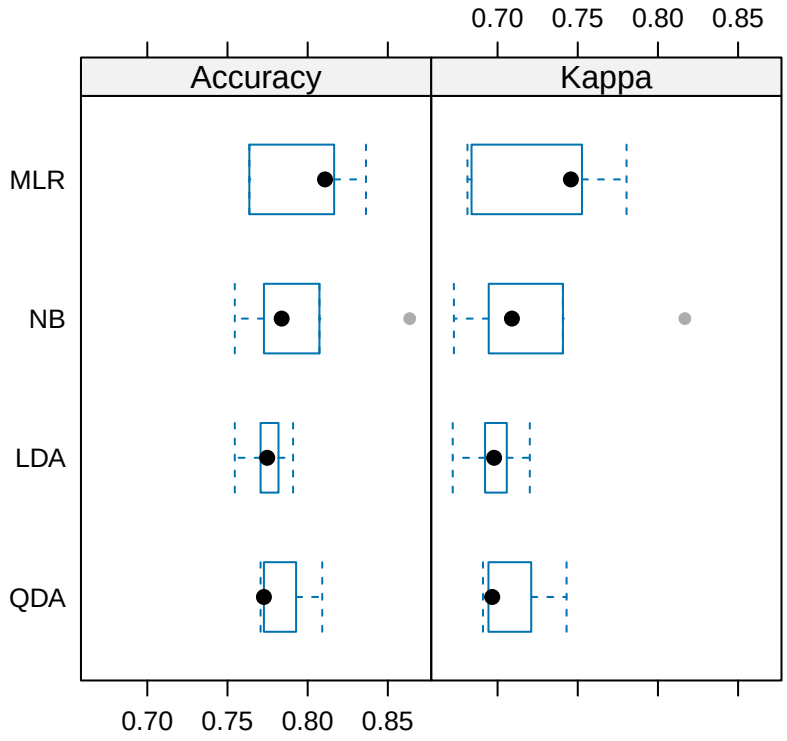

Figure 5: Resampling results using ‘resamples‘ function from the ‘caret‘ package [@caret]

The main results obtained from Fig. 5 can be summarized as follows:
Accuracy:

- Logistic Regression (MLR): Mean accuracy of approximately 79.82%
- Linear Discriminant Analysis (LDA): Mean accuracy of approximately 77.45%
- Quadratic Discriminant Analysis (QDA): Mean accuracy of approximately 78.36%
- Naive Bayes (NB): Mean accuracy of approximately 79.64%

Kappa:

- Logistic Regression (MLR): Mean Kappa of approximately 0.7288
- Linear Discriminant Analysis (LDA): Mean Kappa of approximately 0.6976
- Quadratic Discriminant Analysis (QDA): Mean Kappa of approximately 0.7092
- Naive Bayes (NB): Mean Kappa of approximately 0.7267

Based on these metrics, the logistic regression model (MLR) has the highest mean accuracy and Kappa value among the four models. It is also convenient in terms of interpretability and computational efficiency. Therefore, the logistic regression model may be considered the best model for this classification task.

Feature selection on Multinomial Linear Regression

Lets apply BIC criterion to perform feature selection:

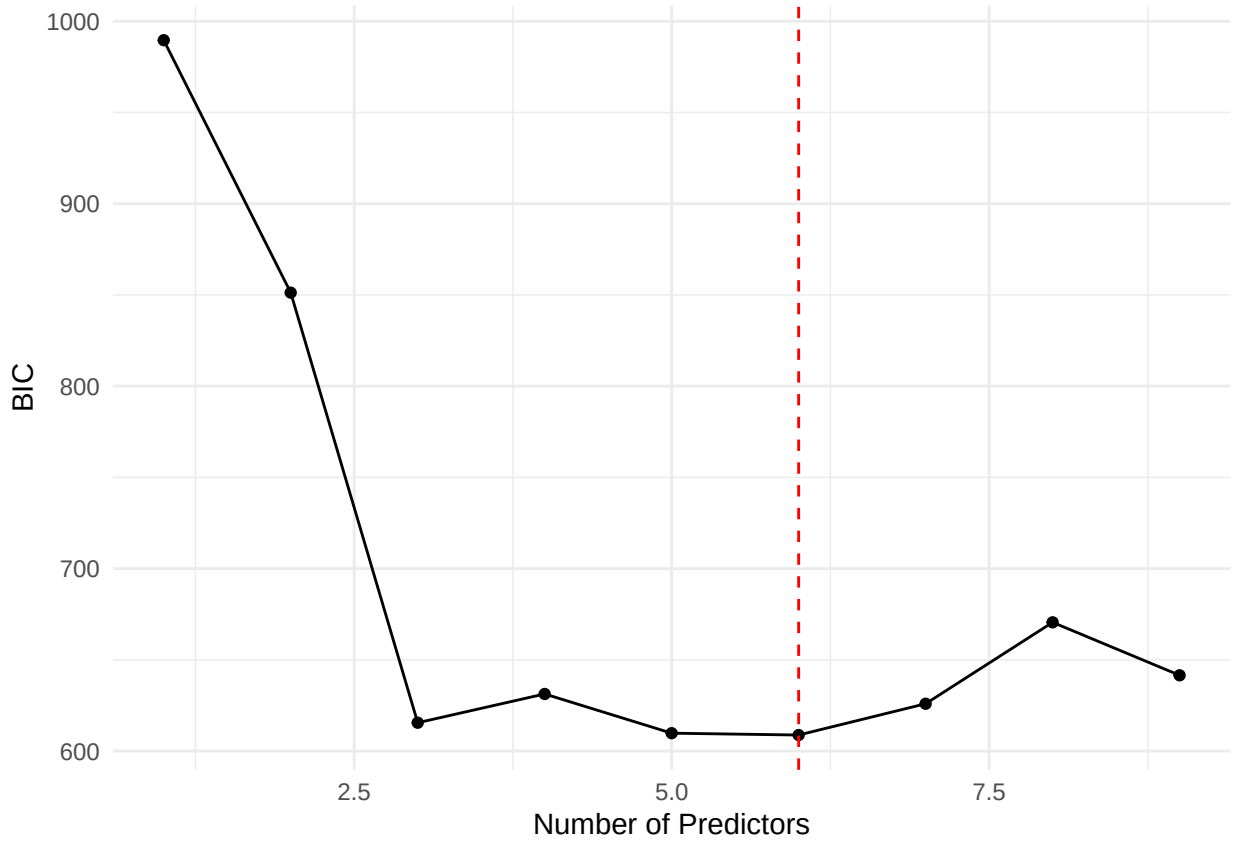

Figure 6: Evolution of BIC with Number of Predictors

The best model found, based in BIC criterion, is `mood ~ acousticness + energy + instrumentalness + liveness + valence + loudness` with a $BIC = 608.80". Fitting this model:

```
## Penalized Multinomial Regression
##
## 550 samples
##   6 predictor
##   4 classes: 'Calm', 'Energetic', 'Happy', 'Sad'
##
## Pre-processing: centered (6), scaled (6)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 440, 440, 440, 439, 441
## Resampling results across tuning parameters:
##
##   decay  Accuracy   Kappa
##   0e+00  0.7982580  0.7286534
##   1e-04  0.7982580  0.7286534
```

```
##   1e-01  0.8001089  0.7311467
##
## Kappa was used to select the optimal model using the largest value.
## The final value used for the model was decay = 0.1.

## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Calm Energetic Happy Sad
##   Calm        38         0     0   2
##   Energetic    0        24     9   0
##   Happy        0         5    17   2
##   Sad          1         1     2  35
##
## Overall Statistics
##
##                Accuracy : 0.8382
##                  95% CI : (0.7654, 0.8958)
##     No Information Rate : 0.2868
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7824
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Calm Class: Energetic Class: Happy Class: Sad
## Sensitivity               0.9744           0.8000       0.6071     0.8974
## Specificity               0.9794           0.9151       0.9352     0.9588
## Pos Pred Value            0.9500           0.7273       0.7083     0.8974
## Neg Pred Value            0.9896           0.9417       0.9018     0.9588
## Prevalence                0.2868           0.2206       0.2059     0.2868
## Detection Rate            0.2794           0.1765       0.1250     0.2574
## Detection Prevalence      0.2941           0.2426       0.1765     0.2868
## Balanced Accuracy         0.9769           0.8575       0.7712     0.9281
```

In this case, since we want to increase the accuracy as much as possible to improve the classification process, we will use all the predictors available to fit the model. Thus, the final model used in our Shiny app will be the following:

```
ctrl <- trainControl(method = "cv", number = 5,
                     classProbs = TRUE,
                     verboseIter=FALSE)

log.fit = train(best_model$formula,
                method = "multinom",
                metric = "Kappa",
```

```
                data = numeric_data,
                preProcess = c("center", "scale"),
                trControl = ctrl,
                trace = FALSE)
# Save the trained model
saveRDS(log.fit, "trained_model.rds")
```

When applying in a future Machine Learning for classification, KNN is expected to outperform LDA and logistic regression in cases where the decision boundary is highly non-linear, given a large n and small p, because KNN is non-parametric. However, accurate classification with KNN requires a substantial number of observations relative to the number of predictors, meaning that n must be much larger. The non-parametric nature of KNN reduces bias but incurs a high level of variance, which can result in a decision boundary that is non-linear when n is moderate or p is not very small. In such situations, QDA may be a more suitable option than KNN as it can provide a non-linear decision boundary while taking advantage of a parametric form. This results in a smaller required sample size for accurate classification compared to KNN.

Shiny app

Now, we can feed our final model to the Shiny app we have built for the user to analyse their playlist and favourite artist. Unfortunately, I have not been able to deploy it on the web yet. The app is located inside the 'mood_app' folder on my GitHub page.

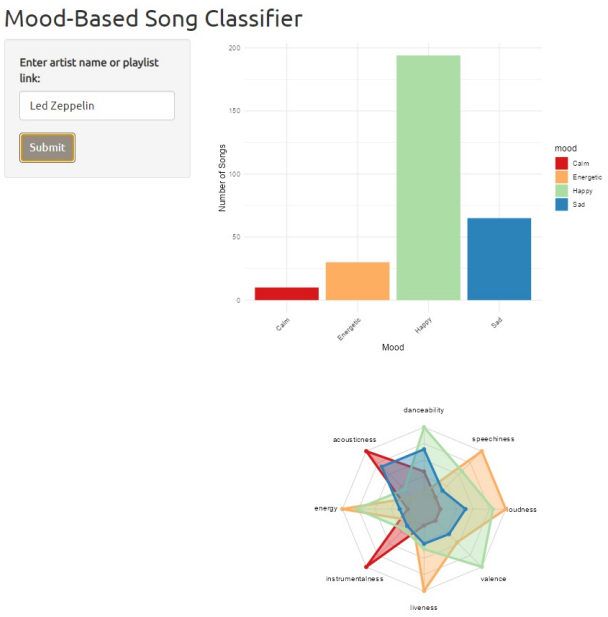

Figure 7: Screenshot of the Shiny app.

References

James, Gareth, Daniela Witten, Trevor Hastie and Robert Tibshirani. 2021. *An Introduction to Statistical Learning: with Applications in R*. Springer.

Kuhn and Max. 2008. "Building Predictive Models in R Using the caret Package." *Journal of Statistical Software* 28(5):1–26.
URL: *https://www.jstatsoft.org/index.php/jss/article/view/v028i05*

Spotify Developer Documentation. N.d. https://developer.spotify.com/documentation/web-api. Accessed: February 19, 2024.

Trochidis, Konstantinos, Grigorios Tsoumakas, George Kalliris and I. Vlahavas. 2008. Multi-label classification of music into emotions. Vol. 2011 pp. 325–330.