Universidade de Brasília
Departamento de Estatística

Trabalho de Conclusão de Curso
Algoritmos De Classificação Aplicados Em Recomendações De Faixas Para *Playlists*.

Rodrigo Cavalcanti Loreto

Projeto apresentado para o Departamento de Estatística da Universidade de Brasília como parte dos requisitos necessários para obtenção do grau de Bacharel em Estatística.

Brasília
2022

Rodrigo Cavalcanti Loreto

Trabalho de Conclusão de Curso
Algoritmos De Classificação Aplicados
Em Recomendações De Faixas Para *Playlists*.

Orientador(a): Prof Leandro Tavares Correia.

Projeto apresentado para o Departamento de Estatística da Universidade de Brasília como parte dos requisitos necessários para obtenção do grau de Bacharel em Estatística.

Brasília
2022

Resumo

Trabalho de conclusão de curso de Estatística com foco em métodos de classificação na análise de faixas do popular serviço de *streaming* Spotify. *Random forest* e regressão logística foram utilizados para identificar características das músicas de *playlists* e diferenciá-las.

Ao longo do estudo ficou evidente algumas vantagens e desvantagens dos métodos. Por exemplo, a facilidade de implementação de ambos, o baixo custo computacional de regressão logística contra a maior demora do *random forest* e o problema de não-convergência na regressão em alguns casos específicos, o que não é um problema possível para *random forest*. Na maioria dos casos, ambos os métodos tiverem desempenhos semelhantes e positivos. Também foram utilizados métodos de validação cruzada: *k-folds* e *leave one out*.

Palavras-chaves: Regressão Logística, Random Forest, K-folds, Leave One Out, playlist, Spotify, classificação, validação cruzada.

Lista de Tabelas

Lista de Figuras

Sumário

1 Introdução

Algoritmos de classificação e recomendação são bastante presentes hoje (CORMEN et al., 2009). Uma das aplicações que provavelmente já cruzou o caminho da maioria das pessoas é a propaganda. No contexto mercadológico de bens e serviços, cada área do mercado possui uma grande gama de subdivisões e cada uma delas possui um público específico. Dado o vasto leque de clientes em potencial, informações sobre usuários são cada vez mais valorizadas. Com elas é possível o direcionamento mais personalizado de produtos, atendendo às especificidades dos nichos, e otimizando custos.

Novas ferramentas nos últimos anos facilitaram a obtenção de dados que possibilitam o mapeamento de perfis, preferências e padrões de navegação dos internautas. Com essas informações e com o desenvolvimento de técnicas computacionais como *data mining*, *machine learning* e técnicas de modelagem estatística, os algoritmos exercem sua função de classificação tanto dos produtos, quanto dos possíveis clientes (WITTEN et al., 2005), (IZBICKI; SANTOS, 2020).

Este trabalho estuda a aplicação de métodos de classificação com foco na indústria da música, mais especificamente nas recomendações de músicas para *playlists* na plataforma de *streaming* Spotify. Antigamente um papel exercido pelos funcionários de lojas de discos e periódicos especializados, hoje esta importante etapa do mercado fonográfico é feita pelos algoritmos. Essa característica dos serviços de música é capaz de definir a escolha do usuário entre plataformas.

Criar listas com sucessos de gêneros específicos como rock, pop e sertanejo pode ser considerado um trabalho simples. Quando se busca uma seleção com foco em temas ou climas específicos, a dificuldade começa a aumentar. Propostas de seleções músicas com foco como “alto astral”, “festa”, “melancólicas”, “descanso”, “correr”, “estudar”, “namorar”, “cozinhar”, entre outras atividades requerem análises mais profundas que discriminem características além do gênero e grau de sucesso da canção.

Pela facilidade de acesso direto aos dados através de uma Interface de Programação de Aplicativos (API) própria, o banco de dados da plataforma Spotify foi utilizado neste estudo.

2 Referencial Teórico

2.1 Regressão Linear

A regressão linear consiste de uma equação para se estimar o valor esperado de uma variável Y (resposta) dados os valores de outras variáveis X (explicativas) (IZBICKI; SANTOS, 2020). É chamada "linear" porque se considera que a relação da resposta às variáveis explicativas é uma função linear de um conjunto de parâmetros. Para se estimar o valor esperado, usa-se uma equação, que determina a relação entre as variáveis

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon. \tag{2.1.1}$$

Onde Y é a variável resposta (dependente), $\beta_j, j = 0, 1, \ldots, p$ são constantes, denominados coeficientes de regressão, $X_j, j = 1, \ldots, p$ são as variáveis explicativas (independentes) e ϵ representa o erro experimental. O parâmetro β_0 corresponde ao intercepto, e fornece a resposta média de Y quando $X_1 = X_2 = \ldots = X_p = 0$. Para $j \geq 1$, os parâmetros β_j indicam uma mudança na resposta média de Y a cada unidade de mudança na variável X_j, quando as demais variáveis são mantidas fixas.

As suposições necessárias para o Modelo de Regressão Linear Múltipla são:

- Os erros não devem ser correlacionados. Devem seguir distribuição normal, ter média zero e variância σ^2 constante. Ou seja, $\epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2)$, para $i = 1, \ldots, n$;
- Deve existir uma relação linear entre a variável dependente e as variáveis independentes;
- Não deve haver multicolinearidade entre as variáveis independentes, ou seja, elas não podem ter correlação alta entre si.

2.2 Regressão Logística

A regressão logística se difere da linear essencialmente pelo fato da variável resposta ser binária, ou seja, Y tem distribuição Bernoulli $(1, \pi)$, com probabilidade de sucesso $P(Y_i = 1) = \pi_i$ e de fracasso $P(Y_i = 0) = (1 - \pi_i)$ (KASSAMBARA, 2018).

No centro da regressão logística está a tarefa de estimar o log odds de um evento. Sendo π a probabilidade de sucesso de um evento, a *odds* de sucesso é definida por:

$$odds = \frac{\pi}{1-\pi}. \tag{2.2.1}$$

A *odds* é de valor não negativo, com valor maior do que 1, quando o sucesso é mais provável de acontecer do que o fracasso. Consequentemente, quando o valor de odds está entre 0 e 1, a probabilidade de fracasso é maior que a probabilidade de sucesso.

Por exemplo, quando $\pi = 0,75$, $1 - \pi = 0,25$ a *odds* de sucesso é $0,75/0,25 = 3$.

Matematicamente, a regressão logística estima uma função de regressão linear múltipla definida por:

$$logit(\pi) = log(\frac{\pi}{1-\pi}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p. \tag{2.2.2}$$

Sendo $\pi = P(Y = 1)$.

Baseado em 2.2.2, chega-se em:

$$\pi = \frac{exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p)}{1 + exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p)} = \frac{e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p}}$$

.

As suposições necessárias para o Modelo de Regressão Logística são:

- A variável dependente precisa ser binária (dicotômica);
- as observações precisam ser independentes umas das outras, ou seja, as observações não devem prover de medições repetidas ou dados correspondentes;
- não deve haver multicolinearidade entre as variáveis independentes, ou seja, elas não podem ter correlação alta entre si;
- deve existir linearidade entre as variáveis independentes e o log odds;
- regressão logística tipicamente requer uma amostra grande.

Para estimar os coeficientes (β) de uma regressão logística, pode-se usar o método de máxima verossimilhança. Nesse caso, dada uma amostra i.i.d. $(X_1, Y_1), \ldots, (X_n, Y_n)$, a função de verossimilhança condicional nas covariáveis, é

$$L(y;(x,\beta)) = \prod_{k=1}^{n}(P(Y_k=1|x_k,\beta))^{y_k}(1-P(Y_k=1|x_k,\beta))^{1-y_k}$$
$$= \prod_{k=1}^{n}\left(\frac{e^{\beta_0+\sum_{i=1}^{d}\beta_i x_{k,i}}}{1+e^{\beta_0+\sum_{i=1}^{d}\beta_i x_{k,i}})^{yk}}\right)\left(\frac{1}{1+e_0^{\beta}+\sum_{i=1}^{d}\beta_i x_{k,i}}\right)^{1-yk}. \quad (2.2.3)$$

Ao se maximizar $L(y;(x,\beta))$, obtém-se estimativas para os coeficientes. Diferente do estimador de mínimos quadrados de uma regressão linear, para maximizar a verossimilhança induzida pela regressão logística, é necessário usar algoritmos numéricos de otimização.

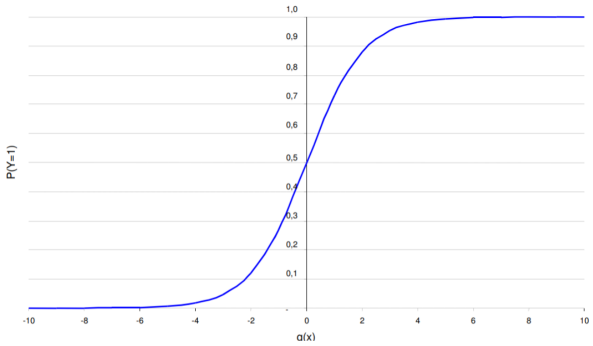

Figura 1: Exemplo de curva de regressão logística.
Fonte:(MIGUEL, 2020)

Como a regressão logística é uma curva, diferente da reta da regressão linear, o coeficiente angular varia ao longo do valor de eixo x pela expressão $\beta\pi(x)[1-\pi(x)]$, aproximando-se de 0 quando $\pi(x)$ se aproxima de 0 ou 1 e atinge o valor máximo quando $\pi(x) = 0,5$.

Por se tratar de um valor de probabilidade, utilizando a Figura 1 de exemplo para ilustrar a ideia, lembrando que Y é uma variável binária, quanto menor o valor de X, maior a probabilidade de $Y = 0$, ou menor a probabilidade de $Y = 1$. Quanto maior, maior a probabilidade de $Y = 1$, e menor a de $Y = 0$.

2.3 Árvores de Decisão

Árvores de decisão consistem em uma metodologia não paramétrica que leva a resultados facilmente interpretáveis. Uma árvore é construída por particionamentos recursivos no espaço das covariáveis. Cada particionamento é chamado de nó e cada resultado final, de folha (RIPLEY, 2007); exemplificado na Figura 2.

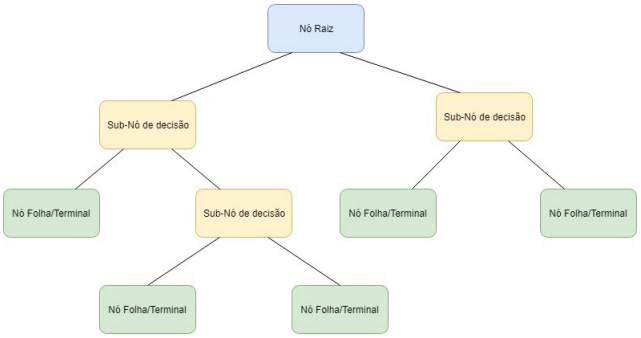

Figura 2: Exemplo de árvore de decisão.
Fonte:(STANKEVIX, 2019)

A forma de utilização da árvore para previsão de uma nova observação é da seguinte maneira: a começar pelo topo, o primeiro nó, verifica-se se a condição descrita é satisfeita. Em caso de sucesso em cumprir a condição, segue-se à esquerda. Caso contrário, à direita. E assim o progresso é feito até o destino de uma folha.

Note como é fácil usar este objeto para entender a relação entre as variáveis explicativas e a variável resposta, ao contrário do que ocorre com a maior parte dos métodos não paramétricos.

Além de serem facilmente interpretáveis, árvores têm a vantagem de lidar trivialmente com covariáveis discretas. Ademais, a maneira como árvores são construídas faz com que covariáveis irrelevantes sejam descartadas. Assim, a seleção de variáveis é feita automaticamente. A estrutura de uma árvore naturalmente lida com interações entre variáveis, ao contrário de modelos lineares, não é necessário incluir termos de interação adicionais.

2.4 Random Forests

O $random forests$ é uma combinação de preditores de árvores tal que cada árvore depende dos valores de um vetor aleatório amostrado de forma independente e com a mesma distribuição para todas as árvores na "floresta". O erro de generalização de uma floresta de árvores de classificação depende basicamente de dois fatores: primeiro, da força das árvores individuais na floresta. Segundo, da correlação entre elas. Esse erro de generalização converge quase certamente para um limite conforme o número de árvores se torna grande. Uma seleção aleatória de observações e de variáveis é utilizada para a criação de cada árvore de classificação. Estimativas internas monitoram o erro, a força e a correlação. Também são usadas para controlar a dinâmica do número de variáveis usadas na separação e para medir a importância entre elas. Todas essas ideias também

são aplicáveis para modelos de regressão com resposta contínua. (BREIMAN, 2001)

Uma vantagem desse método é a prevenção de *overfitting*, que é quando um modelo se ajusta muito bem ao conjunto de dados anteriormente observado, mas se mostra ineficaz para prever novos resultados. Porém, exige um custo computacional maior do que uma árvore de classificação, ou regressão, pois é estimada uma coleção de árvores.

2.5 Avaliação da capacidade preditiva

2.5.1 Matriz de confusão e medidas de acurácia

A Matriz de confusão é uma tabela onde facilmente se identifica todos os quatro tipos de classificação do modelo de classificação binário (isto é, com apenas dois valores distintos na variável resposta). Com ela, facilmente calcula-se valores como acurácia, especificidade, sensibilidade, etc.

Tabela 1: Exemplo de matriz de confusão.

		Resultado da previsão		
		p	**n**	**total**
Classificação verdadeira	**p′**	Verdadeiro Positivo	Falso Negativo	P′
	n′	Falso Positivo	Verdadeiro Negativo	N′
	total	P	N	

A linha p' da matriz denota valor real positivo, n' denota valor real negativo. As colunas p e n denotam valores previstos positivos e negativos respectivamente. P' N', P eN são seus respectivos totais, primeiro das linhas, depois das colunas.

Na tabela acima, a comparação do valor previsto pelo modelo com o valor real de uma observação. Como os nomes sugerem:

- Verdadeiros Positivos (VP) - são observações cujo valor real é positivo e o valor previsto é positivo, isto é, a classificação do modelo é correta.

- Verdadeiros Negativos (VN) - são observações cujo valor real é negativo e o valor previsto é negativo, isto é, a classificação do modelo é correta.
- Falsos Positivos (FP) - são casos em que o resultado correto é negativo entretanto o resultado obtido é positivo, isto é, a classificação do modelo é incorreta.
- Falsos Negativos (FN) - são casos em que o resultado correto é positivo entretanto o resultado obtido é negativo, isto é, a classificação do modelo é incorreta.

Importante evidenciar que na área de aprendizado de máquinas pode-se usar esses 4 quadrantes da matriz de confusão para se obter medidas como:

- Sensibilidade/Recall - S = VP/(VP + FN) (dos casos positivos do evento de interesse, quantos foram corretamente classificados).
- Especificidade - E = VN/(VN + FP) (dos casos negativos do evento de interesse, quantos foram corretamente classificados).
- Valor preditivo positivo/Precision - VPP = VP/(VP + FP) (dos classificados como positivos, quantos foram corretamente classificados).
- Valor preditivo negativo - VPN = VN/(VN + FN) (dos classificados como negativos, quantos foram corretamente classificados).
- Estatística F1 - $F1 = 2/(1/S + 1/VPP)$ (a média harmônica entre Sensibilidade (S) e Valor Preditivo Positivo (VPP))

2.5.2 Data splitting e validação cruzada

Datasplitting e técnicas de validação cruzadas, como $k - folds$ e *leaveoneout* (LOO), existem para tentar evitar super-ajustes, "overfitting", ou simplesmente, ajuste perfeito, dos modelos aos dados utilizados para construí-los.

A primeira ideia consiste em dividir o banco de dados do estudo em 2 partes: treino e validação. Geralmente, essa divisão ocorre com 70% das observações na parte de treino e os 30% restantes, validação. A escolha das amostras que comporão tanto a parte de treino quanto a validação deve, idealmente, ser feita de maneira aleatória. A parte chamada de "treino" vai funcionar exatamente para treinar os modelos. Com essa seção os modelos vão tentar se ajustar da melhor maneira possível aos dados estimando os

valores de seus coeficientes. Então, com a seção de validação, a qual o modelo desconhece, testa-se o desempenho do modelo para classificação das novas observações.

Os métodos $k - folds$ e LOO são elaborados em cima do conceito da partição de dados do *datasplitting*.

$K - folds$ particiona os dados em k lotes. A quantidade de lotes fica a critério do usuário. Após a partição em k lotes, $k - 1$ lotes são utilizados como treinamento e o lote que ficou de fora, validação. O processo se repete até que todos os k lotes tenham sido utilizados como validação.

LOO também particiona o banco em 2 seções, porém, a partição de validação possui apenas 1 observação, deixando todas as outras na partição de treino. Como no $k - folds$, o processo se repete até todas as observações serem usadas como validação.

Além da medida de acurácia também existe o *kappa* de Cohen. Muito útil quando se lida com dados desbalanceados, o *kappa* de Cohen é uma medida de desempenho de modelos de classificação e fica no intervalo $[-1, 1]$. Num exemplo de classificação de observações em 2 classes, -1 quer dizer um desempenho péssimo para ambas as classes, 1, desempenho excelente e 0 ambas as classes são classificadas ao acaso (WIDMANN, 2020).

$$\kappa = \frac{p_0 - p_e}{1 - p_e}. \tag{2.5.1}$$

Sendo p_0 a acurácia do modelo e p_e uma medida de concordância entre a previsão do modelo e o valor real da classe no banco.

No caso de 2 classes:

$$p_e = p_{e1} + p_{e2} = p_{e1,real}p_{e1,prev} + p_{e2,real}p_{e2,prev} \tag{2.5.2}$$

Sendo $p_{ei,real}$ a proporção real de valores da classe i e $p_{ei,pred}$ a proporção prevista pelo modelo na classe i.

3 Metodologia

A linguagem R (TEAM et al., 2013) é utilizada em todo o trabalho.

O objetivo é entender o funcionamento desses algoritmos de classificação e recomendação de músicas, estudar as variáveis criadas pelo Spotify e a relevâncias delas no algoritmo. Criar modelos com métodos de regressão para criar algoritmos de classificação e utilizar dados de *playlists* já criadas para ajustar esses modelos com aprendizado de máquina, e futuramente usar os modelos para criar outras *playlists*.

3.1 API

API é um conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseado na Web. A sigla API refere-se ao termo em inglês *"Application Programming Interface"* que significa, em tradução para o português, "Interface de Programação de Aplicativos". Uma API é criada quando uma empresa de software tem a intenção de que outros criadores de software desenvolvam produtos associados ao seu serviço.

O API do próprio Spotify será utilizado para extrair o banco de dados deste estudo (SPOTIFY, a).

3.2 Conjunto de dados

O banco de dados utilizado será o banco do próprio Spotify, extraído com uso do API da plataforma (SPOTIFY, b). Abaixo, são apresentadas as variáveis presentes no conjunto de dados

- *acousticness* - *Float*. Entre 0 e 1. Representa uma medida percentual de quão acústica é a faixa. 1 representa alta confiança de que ela é acústica, como violões e pianos;
- *analysis_url* - *String*. URL para acessar a análise completa da faixa;
- *danceability* - *Float*. Entre 0 e 1. Descreve quanto apropriada a faixa é para dançar com base em uma combinação de elementos musicais incluindo tempo, estabilidade da batida, força da batida e regularidade geral. 1 sendo o mais dançável;
- *duration_ms* - *Integer*. Duração da faixa em milissegundos;

- *energy* - *Float*. Entre 0 e 1. Representa uma medida percentual entre intensidade e atividade. Tipicamente, faixas energéticas são percebidas como rápidas, barulhentas e de alto volume. Por exemplo, death metal tem alto valor de energia, enquanto que um prelúdio de Bach tem um valor baixo nesta escala. Particularidades perceptíveis que contribuem para essa variável são dinâmica, volume percebido, timbre, ataque e entropia da faixa;

- *id* - *String*. Identificação do Spotify para a faixa;

- *instrumentalness* - *Float*. Entre 0 e 1. Prediz se a faixa não contém voz. "Ooh" e "aah" contam como instrumentos neste contexto. Quanto mais perto de 1, maior a probabilidade da faixa não conter voz. Valores acima de 0.5 tendem a já indicar faixas instrumentais, mas a confiança aumenta quanto mais próximo de 1;

- *key* - *Integer*. Entre 0 e 11. É o tom da música, sendo C, ou Dó, igual a 0, C#/Db=1, D=2... A#/Bb=10(ou "t", ou A) e B=11 (ou "e", ou B);

- *liveness* - *Float*. Entre 0 e 1. Detecta a presença de audiência, ou plateia, na gravação. Quanto maior o valor, maior a probabilidade de ser uma gravação ao vivo. Valores acima de 0.8 indicam alta possibilidade de que é uma gravação ao vivo;

- *loudness* - *Float*. Entre -60 e 0. Volume geral da faixa em decibéis. Valores de *loudness* são calculados pela média da faixa e são úteis para comparar *loudness* relativa entre faixas. *Loudness* é a característica de um som mais correlata com força física (amplitude). Valores tipicamente vão de -60 a 0 db;

- *mode* - *Integer*. 0 ou 1. Indica a modalidade do tom da faixa. 1 é maior e 0 é menor;

- *populatiry* - *Float*. Entre 0 e 100. A popularidade da faixa. Calculada por algoritmo e baseada em grande parte no número total de plays da faixa e quão recentes eles são;

- *speechiness* - *Float*. Entre 0 e 1. Detecta a presença de palavras faladas, não cantadas, na faixa. Quanto mais exclusivamente falada a gravação (talk shows, áudio livros, poesia) mais próximo de 1. Valores acima de 0.66 descrevem faixas que são provavelmente feitas exclusivamente de palavras faladas (declamações, falas, leitura). Valores entre 0.33 e 0.66 descrevem faixas que podem conter palavras faladas e música, tanto em seções quanto sobrepostas. Valores abaixo de 0.33 indicam música e outras faixas que não são de palavras faladas;

- *tempo* - *Float*. Valores positivos. Estimativa geral do andamento de uma faixa em batidas por minuto (BPM). Andamento, no mundo da música, é a velocidade ou andar da peça e deriva diretamente dá média de batidas por minuto;
- *time_signature* - *Integer*. Estimativa de tempo geral da faixa. Tempo é uma notação que diz quantas batidas tem um compasso;
- *track_href* - *String*. Um link para o resultado da busca do API pelos dados da faixa;
- *type* - *String*. Tipo do objeto;
- *uri* - URI do Spotify para a faixa;
- *valence* - *Float*. Entre 0 e 1. Descreve a positividade musical expressa pela faixa. Quanto mais próximo de 1, mais positivo (feliz, alegre, eufórico), quanto mais próximo de 0, mais negativo (triste, deprimido, raivoso).

Das variáveis acima, *danceability*, *valence*, *energy* e *tempo* são classificadas como *"mood"*, clima ou humor. *Loudness*, *speechiness* e *instrumentalness* como *"properties"*, referente as propriedades mais concretas. *Liveness* e *acousticness* como *"context"*, o contexto de performance.

São 19 variáveis para cada música do nosso estudo. *acousticness*, *danceability*, *energy*, *instrumentalness*, *key*, *liveness*, *loudness*, *mode*, *speechiness*, *tempo*, *time_signature* e *valence* são variáveis criadas pelo próprio Spotify com sua análise de cada faixa. A variável *analysis.url* é um link para a análise da faixa.

Para este trabalho o foco é na criação de modelos com base nas variáveis *acousticness*, *danceability*, *duration_ms*, *energy*, *instrumentalness*, *key*, *mode*, *liveness*, *loudness*, *speechiness*, *tempo* e *valence*.

3.2.1 Escolha das *playlists*

As *playlists* escolhidas para este estudo são *playlists* editoriais sem personalização. *Playlists* editoriais são criadas manualmente por editores do Spotify especialistas em gêneros musicais, estilo de vida e cultura, com experiências diversas SPOTIFY (2019). O interesse nesse tipo de *playlist* editorial sem personalização é de evitar qualquer viés que possa estar atribuído ao gosto do usuário.

Começando com um caso de possível dificuldade reduzida, o início das análises teve partida com as *playlists* de maior número de observações e que possuem as propostas

mais distintas entre si. Em seguida, um par de extrema semelhança resultando numa dificuldade bem maior, e um terceiro par de características menos extremas, um meio termo entre os anteriors. Por fim, todos os 3 pares numa mesma análise como último grau de complexidade.

1. Primeiro par - Piano Relaxante e Beast Mode.

   O primeiro par de *playlists* escolhido foi o de maior quantidade de músicas e que aparentou ser o mais distinto. As *playlists* escolhidas para esta primeira análise foram: Piano Relaxante e Beast Mode.

   - Piano Relaxante - "Relaxe com belas peças de piano". Contém apenas peças de piano com o objetivo de relaxar o ouvinte.
   - Beast Mode - "Entre no modo besta". Músicas com o objetivo de animar o ouvinte a prática de exercícios intensos na academia.

2. Segundo par - Alone Again e Life Sucks

   Em seguida, o par que indicou maior semelhança: Alone Again e Life Sucks.

   - Alone Again - "Estar sozinho de novo pode ser difícil, mas essas músicas te farão companhia". Músicas sobre términos de relacionamento e sentir falta de alguém.
   - Life Sucks - "Está tendo um dia ruim? Nós sabemos como é"! Sobre dias, momentos ou situações ruins e difíceis da vida.

3. Terceiro par - Power Hour e Spooning.

   E, como terceiro par, *playlists* de propostas nem muito diferentes nem muito similares. As *playlists* escolhidas foram: Power Hour e Spooning.

   - Power Hour - "Faça *tap back* no *spinning* ou dê uma volta de bicicleta com essas faixas aceleradas". Músicas aceleradas para prática de exercício pedalando.
   - Spooning - "Para momentos aconchegantes". Músicas românticas.

4. Múltiplas *playlists*.

   Por último, o desafio maior. Após as análises dos pares, estudar as 6 *playlists* anteriores em conjunto. Passando pelos mesmos métodos de classificação e validação cruzada.

4 Resultados

4.1 Análise Descritiva

Gráfico de boxplot foi escolhido para iniciar a análise do trabalho. Também foi decidido separar as variáveis em alguns grupos. *Acousticness*, *danceability*, *energy*, *instrumentalness*, *liveness*, *speechiness* e *valence* estão entre o intervalo $[0, 1]$, *loudness* entre $[-50, 0]$, *tempo* (ou BPM) entre $[40, 230)$, *duration* entre $[0, \infty)$.

Dentre as variáveis de intervalo $[0, 1]$, *acousticness*, *instrumentalness*, *liveness* e *speechiness* foram separadas e plotadas como um grupo de boxplots por descreverem a "roupagem" da música. O outro grupo de variáveis de $[0, 1]$, *danceability*, *energy* e *valence*, foram plotadas em outro grupo de boxplots. *Tempo* (BPM), *duration* e *loudness* foram plotadas cada uma individualmente.

4.1.1 Piano Relaxante e Beast Mode

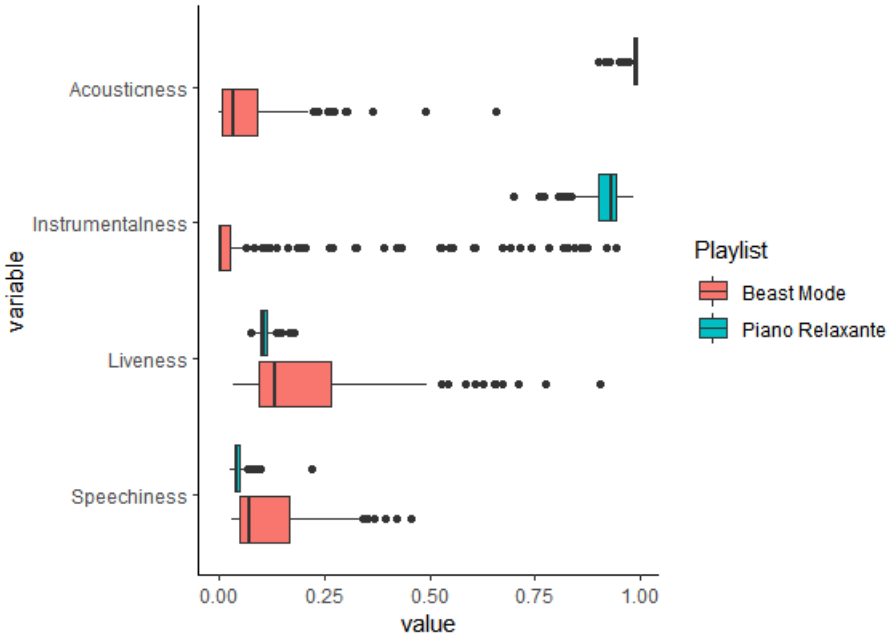

Figura 3: Boxplots do primeiro grupo de variáveis do par Beast Mode e Piano Relaxante.

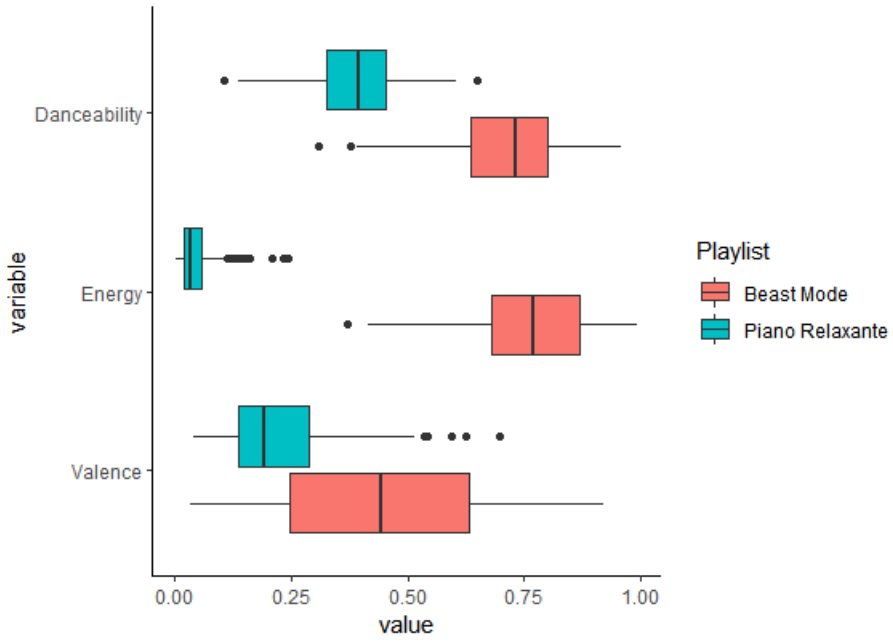

Figura 4: Boxplots do segundo grupo de variáveis do par Beast Mode e Piano Relaxante.

Com base nas Figuras 3 e 4, pode-se visualizar as diferenças entre as músicas e as propostas do primeiro par de *playlists*.

Beast Mode com seu objetivo de animar possui altos valores de *energy* e *danceability*, propondo ao ouvinte músicas animadas, enérgicas e bem ritmadas que te incentivam a gastar energia e manter um ritmo nos exercícios. Já Piano Relaxante possui valores bem baixos de *energy* e valores inferiores de *danceability* em relação à playist anterior pela sua proposta ser justamente o oposto. Em Piano Relaxante o ouvinte procura conservar energia e não há uma procura por ritmos definidos.

Também na segunda *playlist* percebe-se o alto valor de *acousticness*, pelo fato do piano ser um instrumento acústico e único encontrado nas faixas; e *instrumentalness*, acusando a ausência de voz nas músicas, assim como músicas em clínicas de terapia, meditação e outros ambientes que buscam o relaxamento do corpo e mente.

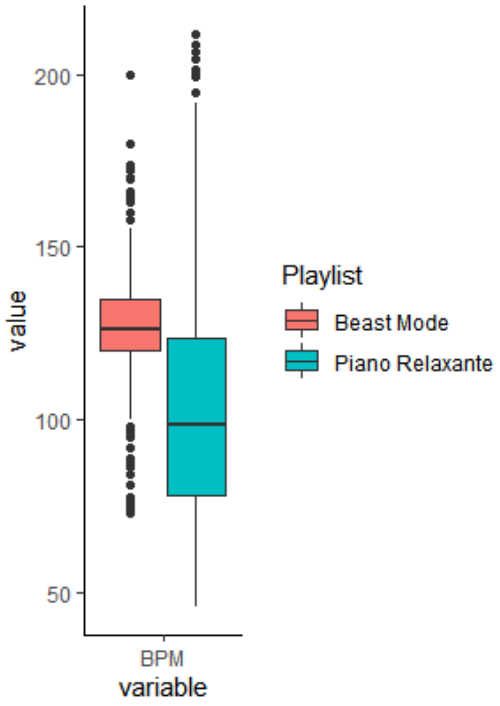

Figura 5: Boxplots da duração do par Beast Mode e Piano Relaxante.

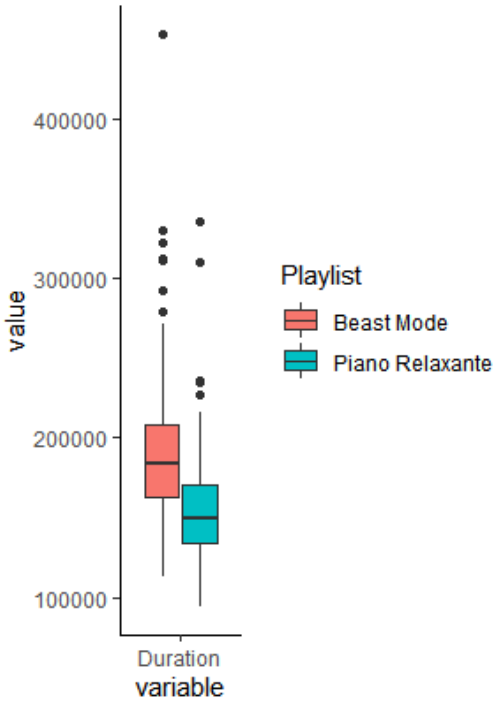

Figura 6: Boxplots do tempo (BPM) do par Beast Mode e Piano Relaxante.

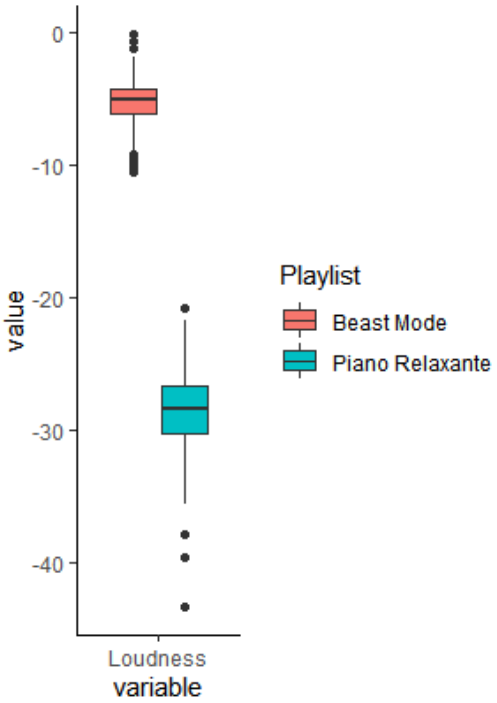

Figura 7: Boxplots de loudness do par Beast Mode e Piano Relaxante.

Analisando as Figuras 5, 6 e 7 tem-se acesso a mais informações que corroboram com a proposta das *playlists*. Antes, viu-se como Piano Relaxante se preocupa menos que suas faixas possuam um ritmo marcado. Agora, através da variável *tempo*, vê-se

uma dispersão grande com músicas de 50 BPM até alguns *outliers* acima de 200 BPM. Essa grande dispersão indica falta de preocupação com um ritmo entre as músicas da *playlist*. Enquanto que Beast Mode possui uma dispersão bem menor. A maioria de suas faixas se encontra entre 100 BPM e 150 BPM, com poucos *outliers* além dessas medidas, justamente pela importância de constância no ritmo para prática de exercícios.

Outra variável interessante de se analisar é *loudness*. Os valores baixos dessa variável (que se trata da amplitude das ondas sonoras e pode ser interpretada como "barulho percebido") pela Piano Relaxante reforçam mais uma vez sua proposta de paz e relaxamento. Por outro lado, Beast Mode procura justamente a tensão com valores altos tendo músicas mais barulhentas. Música alta na academia, música baixa na meditação.

Através dos boxplots identifica-se uma condição particular entre as *playlists*. As variáveis *acousticness*, *energy* e *loudness* separam os dados completamente.

4.1.2 Alone Again e Life Sucks

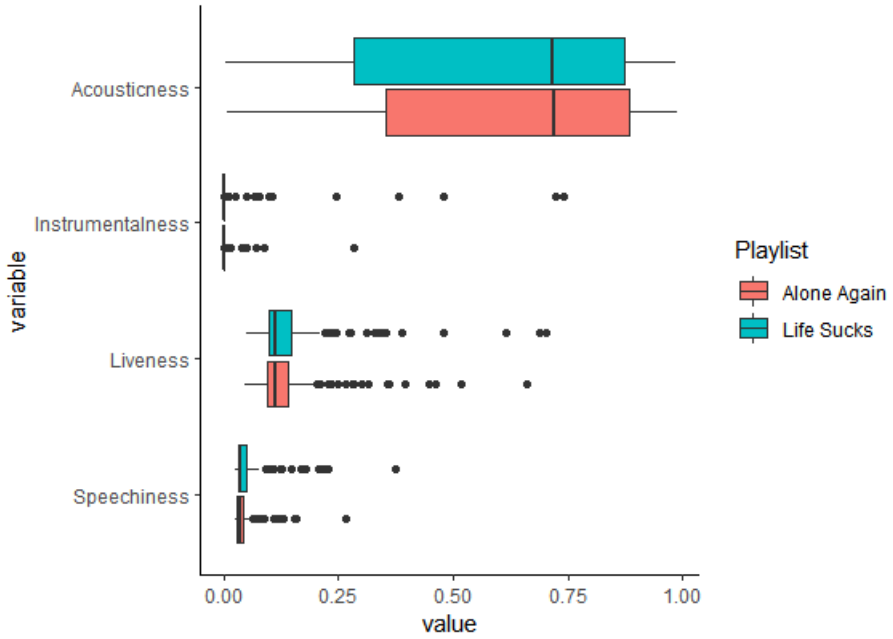

Figura 8: Boxplots do primeiro grupo de variáveis do par Alone Again e Life Sucks.

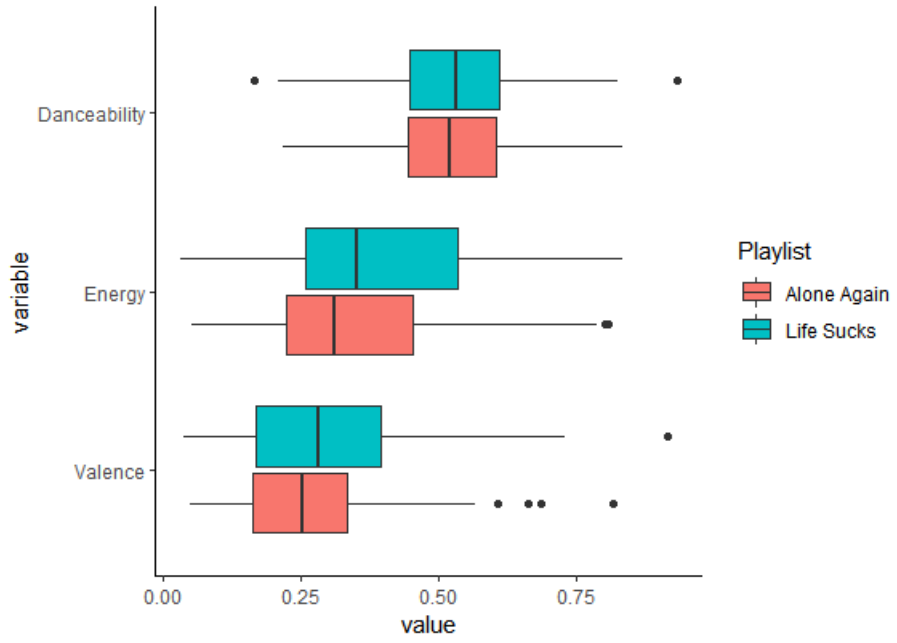

Figura 9: Boxplots do segundo grupo de variáveis do par Alone Again e Life Sucks.

O segundo par é bem diferente do primeiro. Pelo menos é isso que os boxplots nas Figuras 8 e 9 informam. Praticamente idênticas entre todas as variáveis. O que faz muito sentido sabendo que as *playlists* possuem temas muito similares.

Preferência por músicas acústicas, nem muito nem pouco dançantes. Preferência por valores baixos de *energy* e *valence*, corroborando com os climas negativos das duas. E baixíssimos valores de *instrumentalness* e *speechiness* evidenciando a importância da presença de voz com canto, afinal de contas são *playlists* que valorizam a mensagem lírica das músicas.

Inclusive, seria muito interessante a presença de uma variável sobre o conteúdo ou tema poético das faixas, porém ela inexiste no momento de produção deste trabalho.

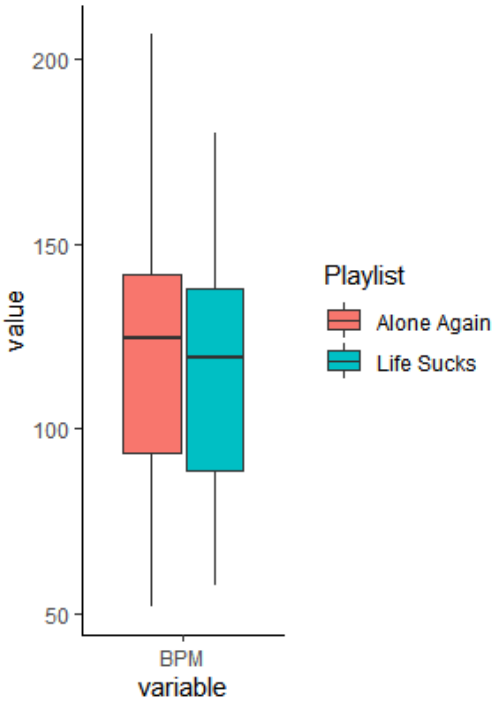

Figura 10: Boxplots da duração do par Alone Again e Life Sucks.

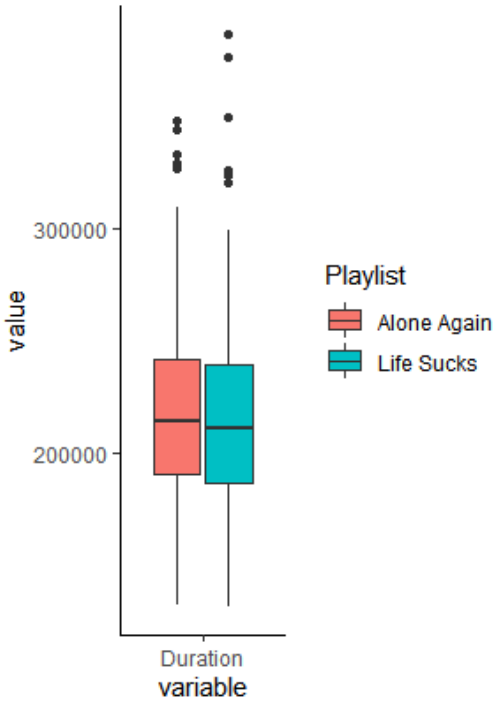

Figura 11: Boxplots do tempo (BPM) do par Alone Again e Life Sucks.

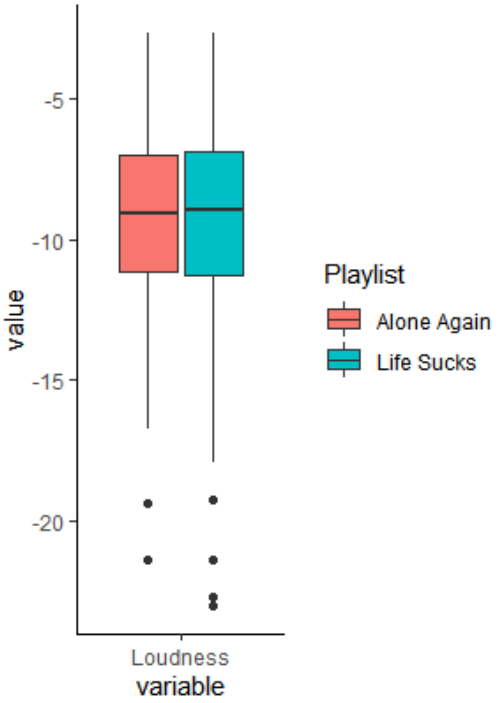

Figura 12: Boxplots de loudness do par Alone Again e Life Sucks.

As Figuras 10, 11 e 12 reforçam a similaridade entre o segundo par. Duração, BPM e volume possuem diferenças mínimas, estando a maior delas na dispersão da variável *tempo*. Nela, Alone Again possui algumas poucas faixas com valores mais ex-

tremos, próximo e superiores a 200 BPM e próximos de 50 BPM. Fora isso, são praticamente idênticas: muito acústicas, poucas ou nenhuma faixa instrumental, muito calcadas nas letras, maioria das faixas com clima negativo ou de tristeza e pouco enérgicas.

Os métodos terão muita dificuldade em diferenciar este par. Possivelmente a variável inexistente sobre o tema da letra da música seria determinante neste caso.

4.1.3 Power Hour e Spooning

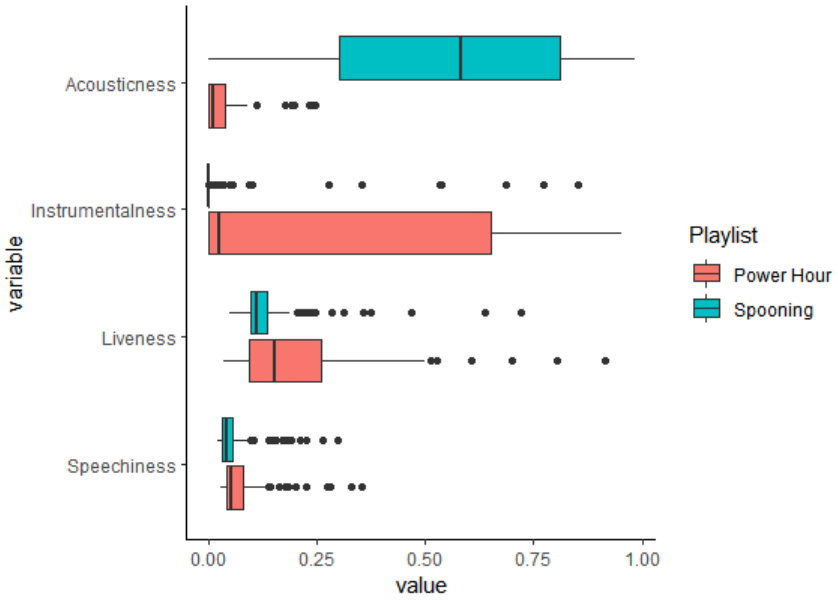

Figura 13: Boxplots do primeiro grupo de variáveis do par Power Hour e Spooning.

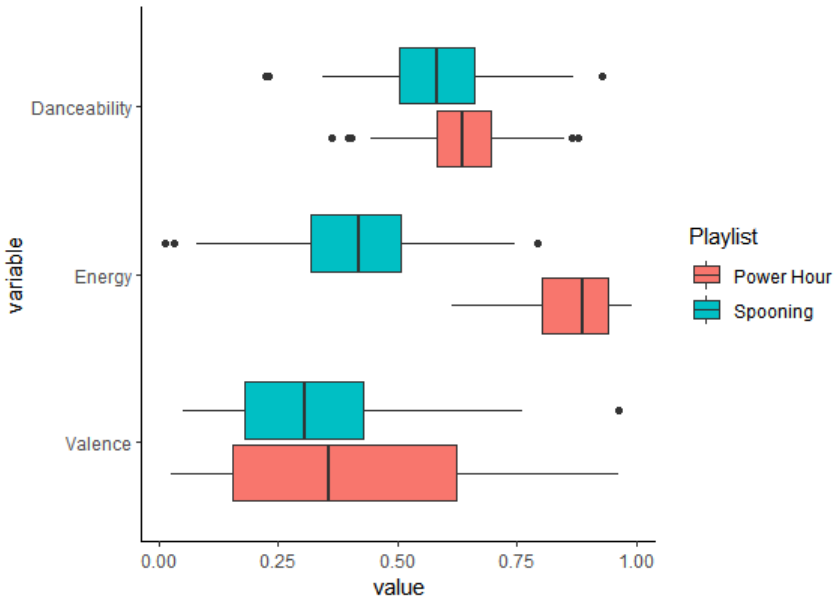

Figura 14: Boxplots do segundo grupo de variáveis do par Power Hour e Spooning.

Agora, o terceiro e último par. Utilizando as Figuras 13 e 14 vê-se as diferenças e semelhanças entre as *playlists*, suas propostas e climas.

Spooning com altos valores para *acousticness* indica forte presença de instrumentos acústico como violões e pianos, comumente utilizados em músicas mais intimistas e românticas, justamente a proposta da *playlists*. Baixíssimos valores para *instrumentalness* e *speechiness* evidenciam a pequena quantidade, ou ausência, de faixas instrumentais. Por consequência, isso reforça a presença de canto nas músicas, outro fator importante para o clima da *playlist*.

Power Hour possui valores baixíssimos de *acousticness* e altíssimos para *energy*, diferenciando-se bastante do seu par com faixas muito enérgicas e que utilizam instrumentos diferentes, praticamente com ausência de instrumentos acústicos. Ambas as características são fortes para uma seleção de músicas com foco em exercício físico.

Além disso, a dispersão grande em *instrumentalness* mostra a falta de preocupação com a presença ou não de voz nas faixas de Power Hour, enquanto que é algo muito importante para Spooning.

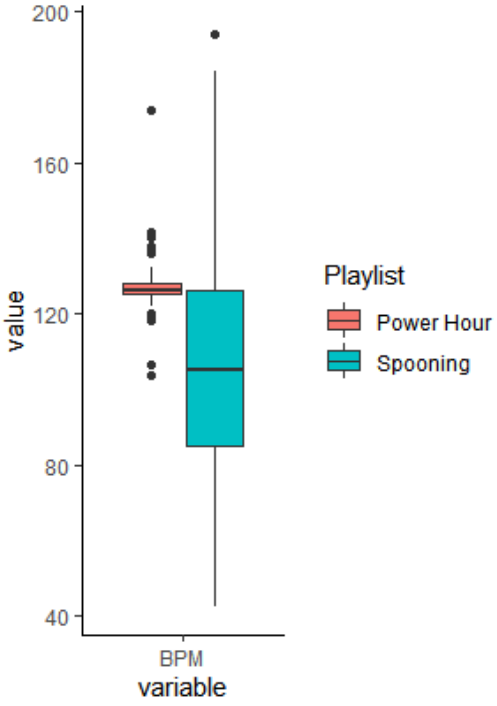

Figura 15: Boxplots da duração do par Power Hour e Spooning.

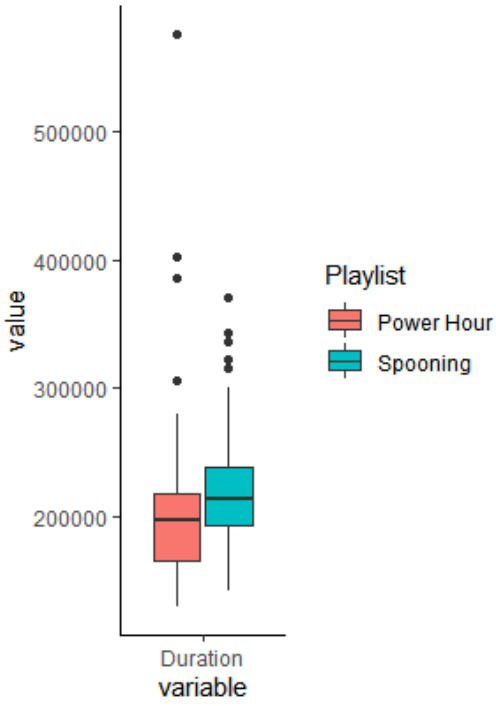

Figura 16: Boxplots do tempo (BPM) do par Power Hour e Spooning.

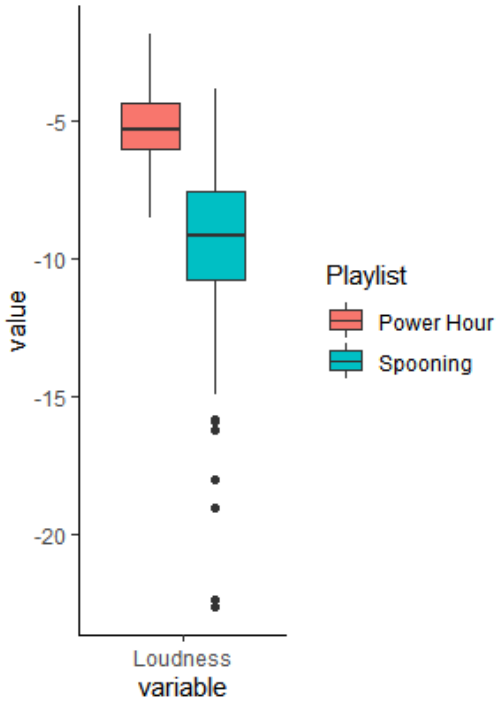

Figura 17: Boxplots de loudness do par Power Hour e Spooning.

As diferenças de propostas também estão presentes nas Figuras 15, 16 e 17. Especialmente em 15 com a alta densidade de faixas entre 120 BPM e 130 BPM, homogeneidade que reforça a importância de músicas com *tempo* próximo para a manutenção de um ritmo durante exercícios; e 17 que endossa maiores valores de volume para a prática de atividade física. Enquanto que Spooning não aparenta ter preocupação com regularidade de *tempo*, apresentando valores quase tão baixos quanto 40 BPM e chegando próximo de 200 BPM. Spooning também não demonstra se preocupar muito com um homogeneidade de volume em 17 mas é notada uma preferência por volumes mais baixos, inclusive, com uma observação inferior a $-20db$.

Com respeito a grau de dificuldade para os métodos, o terceiro par está num certo meio termo entre os pares passados.

4.1.4 Múltiplas *playlists* juntas

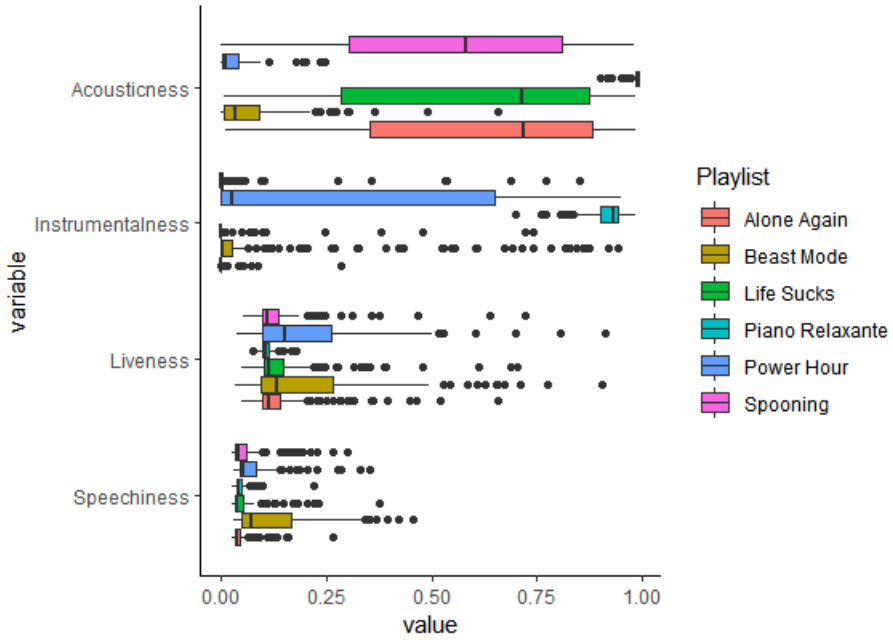

Figura 18: Boxplots do primeiro grupo de variáveis de todas as *playlists*.

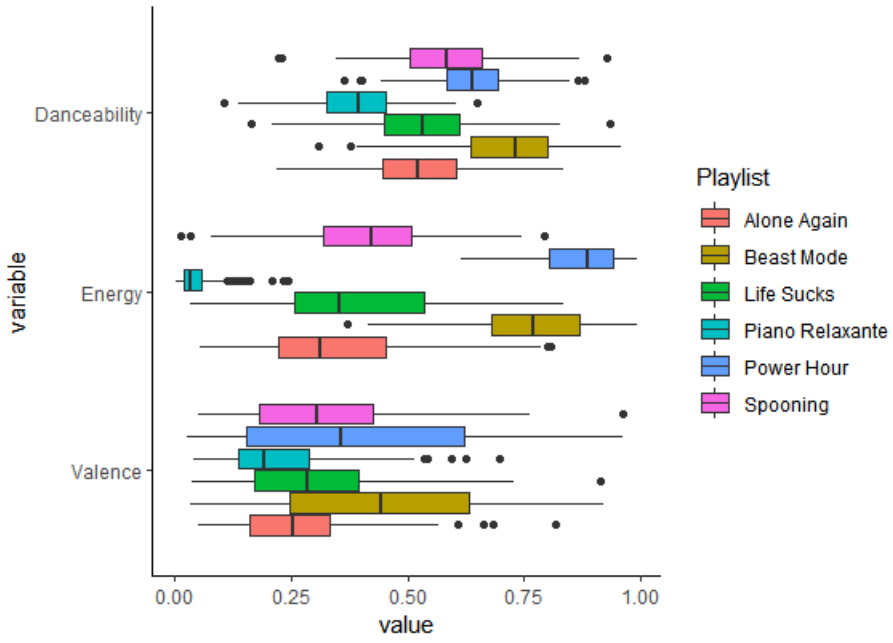

Figura 19: Boxplots do segundo grupo de variáveis de todas as *playlists*.

Analisando os boxplots das Figuras 18 e 19 são vistos alguns comportamentos semelhantes entre *playlists*. Alone Again, Life Sucks e Spooning são muito parecidas em todas as variáveis. Power Hour e Beast Mode também são muito similares. Diferem na dispersão apenas em *instrumentalness*. E Piano Relaxante está praticamente isolada em

acousticness, *instrumentalness* e *energy*.

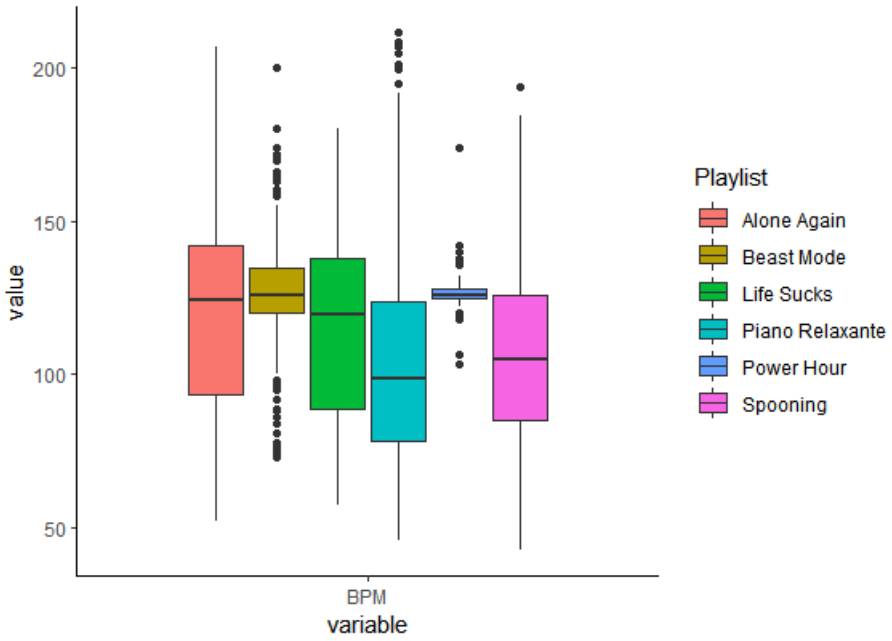

Figura 20: Boxplots da duração de todas as *playlists*.

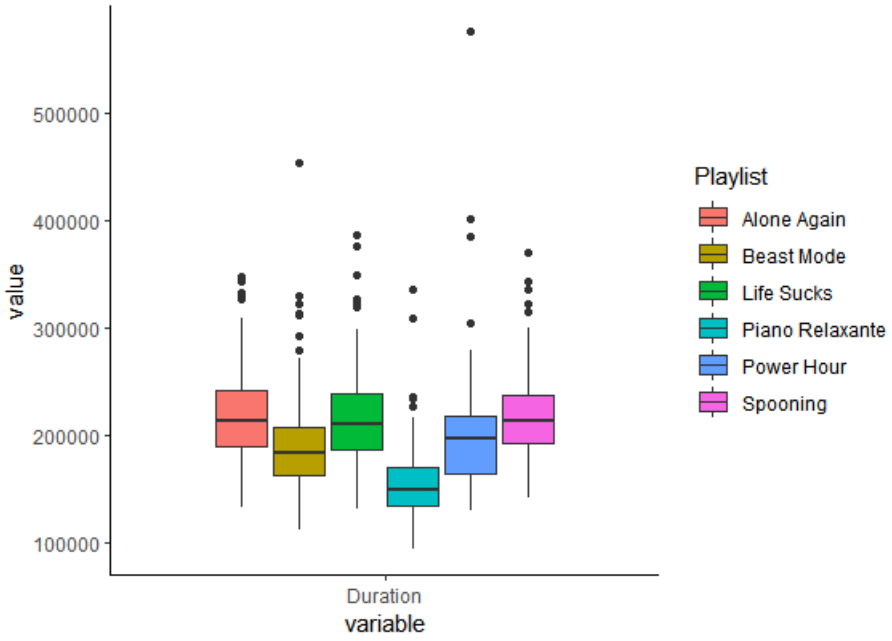

Figura 21: Boxplots do tempo (BPM) de todas as *playlists*.

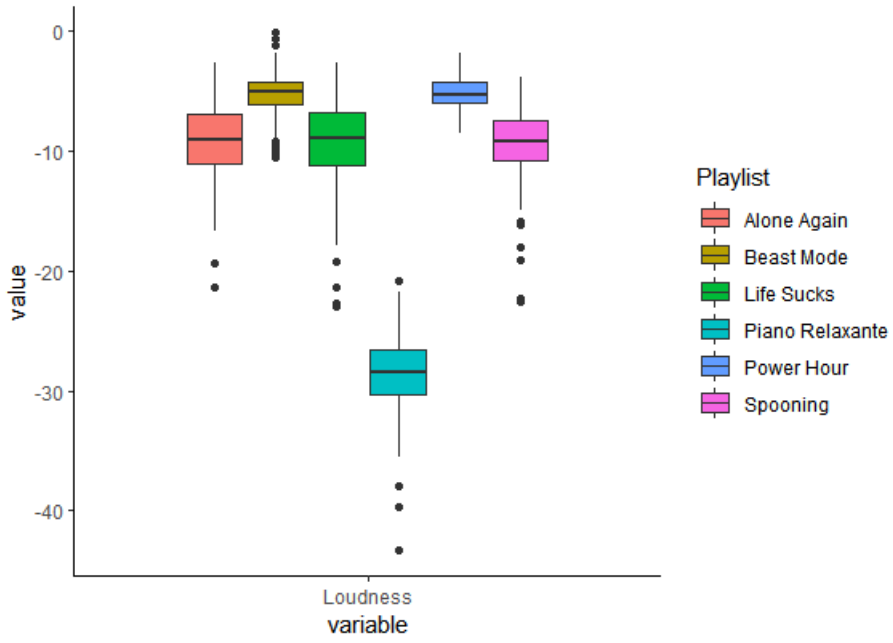

Figura 22: Boxplots de loudness de todas as *playlists*.

Com base nos boxplots restantes, Figuras 20,21 e 22, constata-se que a semelhança de Alone Again, Life Sucks e Spooning persiste. Beast Mode e Power diferem na dispersão de *tempo*. E Piano Relaxante se afasta um pouco das demais *playlists* em *duration* e isola quase completamente em *loudness*.

Tem-se aqui, então, praticamente 3 grupos de *playlists*:

1. Alone Again, Life Sucks e Spooning.
2. Beast Mode e Power Hour.
3. Piano Relaxante.

O grupo 1 com valores altos para *acousticness*, *danceability* e *loudness*. Baixos para *energy* e *valence*. Baixíssimos para *instrumentalness*, *liveness* e *speechiness*. E os valores mais altos de *duration*.

Grupo 2 com valores altíssimos para *loudness* e *energy*. Altos para *danceability*. Baixos para *valence*. Baixíssimos de *acousticness*, *liveness* e *speechiness*. Também interessante notar que possui a menor dispersão na variável *tempo*.

Grupo 3 com valores altíssimos para *acousticness* e *instrumentalness*. Baixos para *danceability*. Baixíssimos para *liveness*, *speechiness*, *energy*, *valence* e *loudness*. Sendo também o grupo de menores valores de *duration* e a maior dispersão de *tempo*.

Com essas fortes características em comum de cada grupo, é esperado que os métodos tenham bastante dificuldade em separar as *playlists* do mesmo grupo.

No caso do grupo 1, é possível que, caso existisse uma variável que trabalhasse com o conteúdo das letras das músicas, ela seria peça importante para a diferenciação dentro do grupo.

O grupo 2 já é um caso diferente. Suas *playlists* são focadas na prática de atividades físicas, ou seja, seus critérios para a inclusão de músicas é idêntico, ou muito similar, tornando as músicas presentes nesses *playlists* praticamente intercambiáveis.

O grupo 3 só possui 1 *playlist* e se diferencia bastante dos outros grupos.

Interessante notar também que a variável *energy* representada na Figura 19 indica uma boa separação entre os grupos. Grupo 1 com valores intermediários, grupo 2 com valores altíssimos e grupo 3 com valores baixíssimos.

4.2 Modelagem Beast Mode vs Piano Relaxante

O início da modelagem é com o método de regressão logística e depois passa-se pelo método de *random forest*. Dentro de todos os métodos o primeiro modelo utiliza as variáveis *acousticness*, *danceability*, *duration*, *energy*, *instrumentalness*, *key*, *liveness*, *loudness*, *mode*, *speechiness*, *tempo* e *valence* para tomar a decisão de classificar as observações do banco. Por fim, ambos os métodos são trabalhados pelas validações cruzadas *k-folds* e *leave one out* (abreviada para *LOO*). Neste primeiro par, a decisão é entre Beast Mode (0) e Piano Relaxante (1).

Entretanto, como suspeitado pela análise dos boxplots, este modelo não converge. Com isso a regressão logística produz estimadores com alta variabilidade trazendo instabilidade às estimativas dos paramêtros. Isso pode ocorrer pela separação completa, ou quase completa, dos dados por parte de alguma variável.

No primeiro par existe não só 1, mas 3 variáveis que ocasionam isso. São elas: *acousticness*, *energy* e *loudness*. Também ocorre uma separação quase-perfeita na variável *instrumentalness*. Graças a essas separações, seu treinamento e previsão têm resultados de classificação perfeitos.

Pelo método de *random forest*, apesar dele não ser suscetível a "não convergência", a separação completa também faz com que o modelo produza resultados perfeitos, tanto no treino quanto nas previsões. Como não consta no trabalho tal modelo pelo método de regressão logística, também não o utilizamos com *random forest*. E assim ele também

não passará pelas validações cruzadas.

Para os algoritmos de classificação essas 4 variáveis serão desconsideradas dos modelos.

4.2.1 Modelo 1 - Treino e previsão

Utilizando as variáveis *danceability*, *duration*, *key*, *liveness*, *mode*, *speechiness*, *tempo* e *valence*.

- Treino - Regressão logística

A regressão indica *danceability* como variável mais significativa para o modelo, de acordo com a análise do boxplot na Figura 4. Faz sentido essa indicação, visto que das variáveis que restam, é a que melhor separa os dados e também é a que mais contribui com este modelo.

Tabela 2: Matriz de confusão regressão logística do modelo 1.

	Beast Mode	Piano Relaxante
Beast Mode	138	3
Piano Relaxante	3	245

Seu treino tem resultado quase perfeito com erro de apenas $1,54\%$. Este resultado mostra como a separação entre as *playlists* ainda é muito clara com as variáveis restantes.

- Treino - *Random forest*.

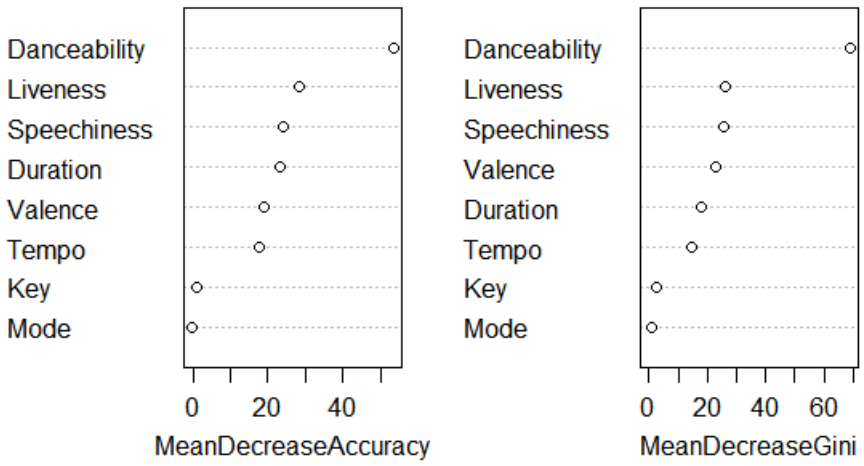

Figura 23: Importância de variáveis para o modelo 1.

Assim como no modelo de regressão, *danceability* é indicada como a variável mais significativa, enquanto que *key* e *mode* as menos. Novamente um resultado que faz bastante sentido. Tom e modalidade não são características marcantes para a proposta de qualquer das duas *playlists* deste par.

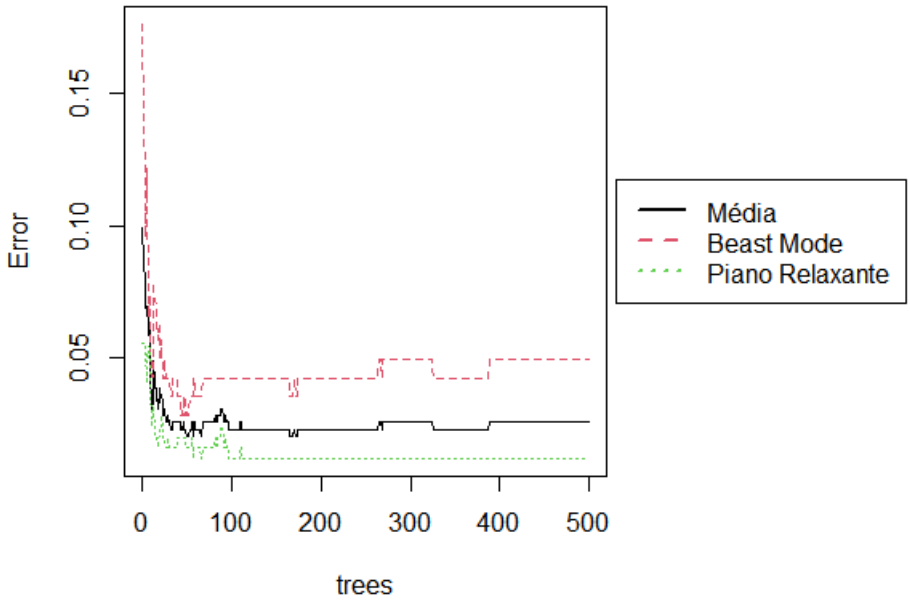

Figura 24: Erro do treino do *random forest* para o modelo 1.

Pela Figura 24 vemos o comportamento do erro do treino e a importância do cuidado com a quantidade de árvores no método de *random forest*. A grosso modo, quanto mais árvores, menor e mais estável o erro médio. Porém, o excesso de árvores pode causar o superajuste. Percebe-se um decrescimento rápido do erro antes mesmo de 100 árvores, porém, após aproximadamente 250 árvores o erro torna a crescer. Curiosamente, após se estabilizar próximo da quantida de 100 árvores, a *playlist* Piano Relaxante tem um erro constante muito próximo de 0 e apenas Beast Mode segue tendo alterações.

Tabela 3: Matriz de confusão *random forest* do modelo 1.

	Beast Mode	Piano Relaxante
Beast Mode	134	3
Piano Relaxante	7	245
class.error	0.05	0.01

Com resultado próximo à regressão logística, o modelo performa quase perfeitamente com pequeno erro de $2,57\%$.

- Previsões

Tabela 4: Previsão regressão logística modelo 1.

	Beast Mode	Piano Relaxante
Beast Mode	58	2
Piano Relaxante	1	97

Tabela 5: Previsão *random forest* modelo 1.

	Beast Mode	Piano Relaxante
Beast Mode	58	1
Piano Relaxante	1	98

Comparando o desempenho das duas previsões, temos em 4 o erro de $1,9\%$ para o método de regressão logística e em 5 o erro de $1,27\%$ para *random forest*. Ambos os métodos tiveram resultados excelentes de quase perfeição. Curiosamente o método que obteve o pior resultado de treinamento performou a melhor previsão.

4.2.2 Modelo 2 - Treino e previsão

Utilizando *danceability*, *duration*, *liveness*, *speechiness* e *tempo*.

- Treino - Regressão logística

Danceability continua como variável mais significativa do modelo.

Tabela 6: Matriz de confusão regressão logística do modelo 2.

	Beast Mode	Piano Relaxante
Beast Mode	137	3
Piano Relaxante	4	245

Visto que *danceability* também está presente neste modelo, não é surpresa outro resultado bastante positivo. Das que permanecem, é a variável que melhor representa a proposta de cada *playlist*. Treinamento tem classificação quase perfeita com erro de $1,8\%$.

- Treino - *Random forest*

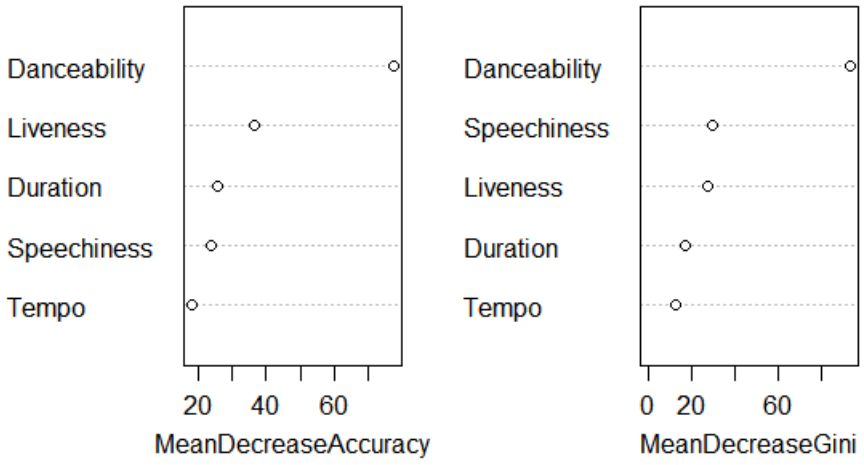

Figura 25: Importância de variáveis para o modelo 2.

A variável mais importante segue sendo *danceability*, assim como no método de regressão linear.

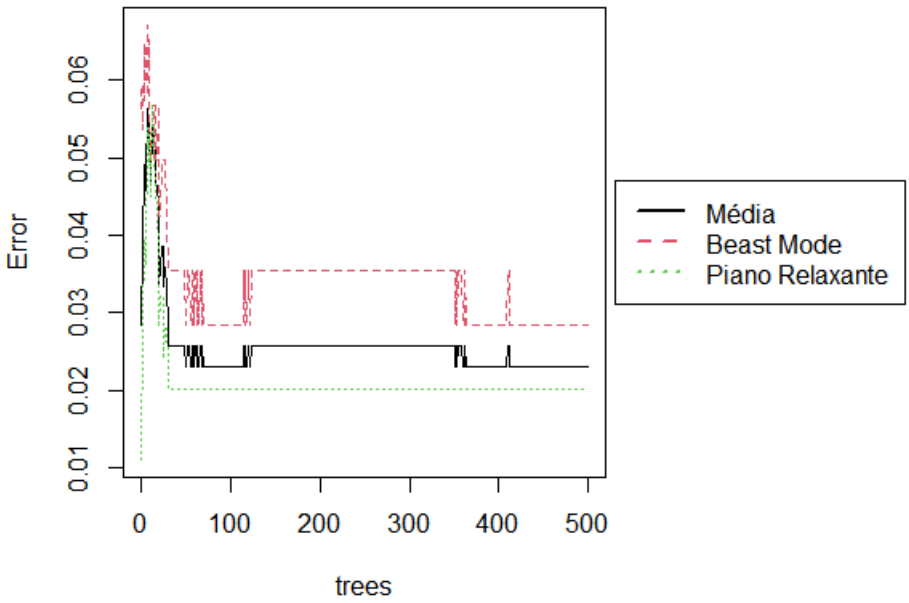

Figura 26: Erro do treino do *random forest* para o modelo 2.

Na Figura 26 o gráfico de erro demonstra uma aproximação entre os erros de cada *playlist*. Novamente Piano Relaxante fica com erro estagnado enquanto que Beast Mode varia.

Tabela 7: Matriz de confusão *random forest* do modelo 2.

	Beast Mode	Piano Relaxante
Beast Mode	137	5
Piano Relaxante	4	243
class.error	0.03	0.02

Resultado de treino muito bom, quase perfeito. Erro de $2,31\%$.

- Previsões

Tabela 8: Previsão regressão logística modelo 2.

	Beast Mode	Piano Relaxante
Beast Mode	58	2
Piano Relaxante	1	97

Tabela 9: Previsão *random forest* modelo 2.

	Beast Mode	Piano Relaxante
Beast Mode	58	2
Piano Relaxante	1	97

Desta vez o desempenho de ambos os métodos foi idêntico, mesmo após *random forest* ter tido um treino com erro pior. Inclusive, seu erro na previsão foi melhor do que o resultado do treino, enquanto que o método de regressão teve um desempenho melhor no treino do que na previsão. Novamente as previsões são muito boas. Quase perfeitas. Corroborando com a capacidade dos métodos em distinguir as *playlists* em estudo. Ambas tiveram erro de apenas $1,9\%$.

4.2.3 Comparação entre métodos - Beast Mode vs Piano Relaxante

Reforçando o que foi visto nos boxplots, este primeiro par é bem diferente entre si. Além disso, Piano Relaxante é bem homogênea, ocasionando o rápido "aprendizado" por parte dos modelos, como vimos nas Figuras 24 e 26. Os excelentes resultados, tanto no treino quanto nas previsões, em ambos os métodos, mostra como as diferenças entre as *playlists* estão bem descritas nas variáveis disponíveis, principalmente em *danceability*, apontada por ambos os métodos como variável mais significativa. E, de fato, pelo ponto de vista da proposta das *playlists* em cheque, um ritmo marcado e forte é muito requisitado numa *playlist* de exercícios físicos, e preterido por um ritmo mais brando para relaxar.

4.3 Validação cruzada

Tabela 10: Melhores resultados validação cruzada por método.

CV	Método	Acurácia	Kappa
K-folds	Regressão logística	0.9781	0.9522
K-folds	*Random forest*	0.9763	0.9488
LOO	Regressão logística	0.9799	0.9566
LOO	*Random forest*	0.9781	0.9527

No geral, todos os modelos dos métodos performaram muito bem com uma taxa de acerto quase perfeita. O primeiro par serviu bem seu papel de "par fácil" e mostrou,

para as variáveis disponíveis a análise, elas não só são *playlist* diferentes, são também fáceis de diferenciar.

Pelo *k-folds* o modelo 2 de regressão logística e pelo *LOO*, um empate técnico entre o modelo 2 de ambos os métodos. Sendo assim, podemos dizer que as duas validações concordam que o modelo 2 do método de regressão logística possui o melhor desempenho. Curioso notar que a melhor previsão feita antes da validação cruzada foi feita pelo modelo 1 de *random forest*, modelo que não foi apontado pelas validações como melhor. Esse pode ser considerado um caso de "sorte" na amostra selecionada pelo *data splitting*.

4.4 Modelagem Alone Again vs Life Sucks

Diferente do par passado, as *playlists* do segundo par são bem mais similares. Não encontramos variáveis que causem a separação completa ou quase completa dos dados e assim não temos casos de não convergência pela regressão logística. Desta vez classificamos as faixas entre Alone Again (0) e Life Sucks (1).

4.4.1 Modelo 1 - Treino e previsão

Começamos o estudo do atual caso com o modelo utilizando as variáveis *acousticness*, *danceability*, *duration*, *energy*, *instrumentalness*, *key*, *liveness*, *loudness*, *mode*, *speechiness*, *tempo* e *valence*.

- Treino - Regressão logística

Como visto nos boxplots nas Figuras 8, 9, 10, 11 e 12, estas *playlists* são muito similares em todas as variáveis, sintoma da similaridade de seus temas e climas. O modelo de regressão aponta apenas 2 variáveis como significativas para a classificação. São elas *energy* e *loudness*. Analisando novamente os boxplots, **energy** parece ser a variáveis que mais difere as *playlists*, mas mesmo assim são dispersões muito parecidas.

Tabela 11: Matriz de confusão regressão logística do modelo 1.

	Alone Again	Life Sucks
Alone Again	46	27
Life Sucks	56	116

Pela Tabela 11 vemos a dificuldade substancialmente maior deste par. As faixas

de Alone Again estão quase que divididas meio a meio. Temos mais previsões erradas para esta *playlist* do que corretas. O resultado de 46 faixas $(45,1\%)$ classificadas corretamente e 56 $(54,9\%)$ incorretamente é muito próximo da proporção de faixas de cada *playlist* no estudo. $41,63\%$ das observações neste treino são Alone Again e $58,37\%$, Life Sucks. Um resultado tão próximo do acaso é um forte indicativo de incapacidade de distinção entre *playlists*. Entreanto, Life Sucks possui um resultado bem melhor. O treino tem erro de $33,88\%$.

- Treino - *random forest*

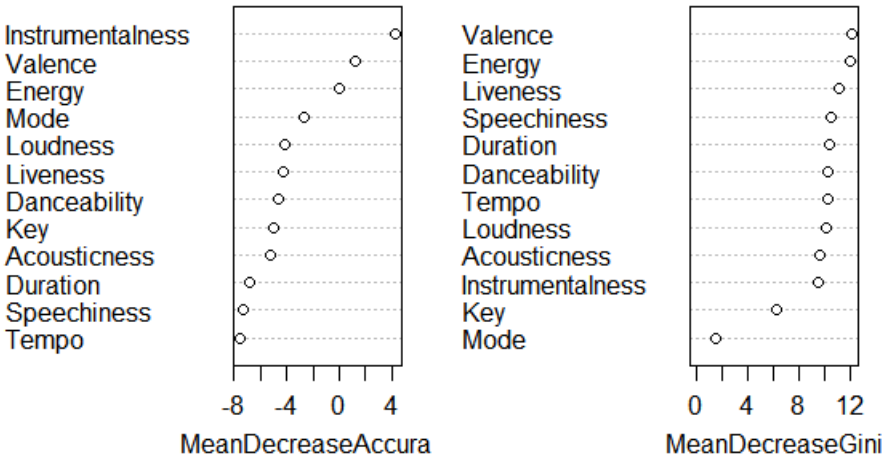

Figura 27: Importância de variáveis para o modelo 1.

Agora estudando o par através do *random forest* nota-se que as variáveis mais importantes indicadas são *instrumentalness*, *valence* e *energy*. Os métodos apenas concordam com a importância de *energy*.

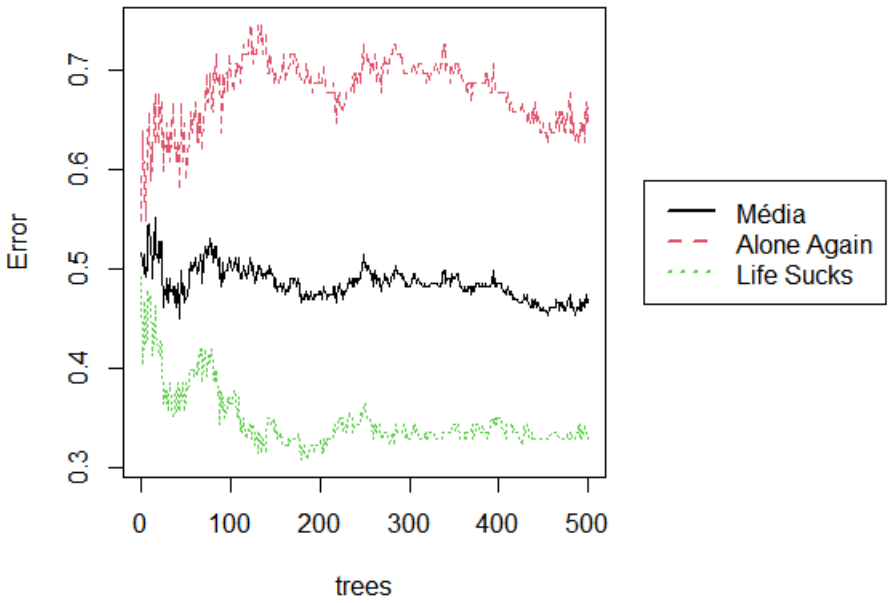

Figura 28: Erro do treino do *random forest* para o modelo 1.

Pela Figura 28 evidencia-se um comportamento semelhante ao treino de regressão logística. A maior dificuldade está na classificação correta da *playlist* Alone Again. Com o aumento do número de árvores ela passa a ter uma melhora, mas o erro é muito alto. No número final de árvores, o modelo tem quase o dobro de erro para classificar observações Alone Again em relação a Life Sucks.

Tabela 12: Matriz de confusão *random forest* do modelo 1.

	Alone Again	Life Sucks
Alone Again	36	48
Life Sucks	66	95
class.error	0.65	0.34

O resultado do treino apresenta a grande dificuldade na classificação das observações. Ambos os métodos tiveram muita dificuldade no treino. Erro de $46,53\%$ muito próximo de 50% indicando a forte possibilidade de classificações feitas ao acaso, reforçando a grande similaridade entre as *playlists* disposta pelas variáveis disponíveis e suas propostas semelhantes.

- Previsões

Tabela 13: Previsão regressão logística modelo 1.

	Alone Again	Life Sucks
Alone Again	12	18
Life Sucks	36	39

Tabela 14: Previsão *random forest* modelo 1.

	Alone Again	Life Sucks
Alone Again	12	23
Life Sucks	36	34

Na etapa de previsões vemos resultados muito ruins. Previsões bem parecidas entre si e bem mais à vontade para classificar as faixas como pertencentes a Life Sucks. Com erros de $51,43\%$ para regressão e $56,19\%$ para *random forest* ambas os métodos erram mais do que acertam, mas a maior decepção está na regressão logística que teve treino ruim mas bem melhor do que seu concorrente. Os pares se mostram muito semelhantes para serem bem diferenciados com base nas variáveis disponíveis aos métodos.

4.4.2 Modelo 2 - Treino e previsão

Utilizando as variáveis *energy* e *loudness*.

- Treino - Regressão logística

Tabela 15: Matriz de confusão do treino do modelo 2.

	Alone Again	Life Sucks
Alone Again	30	20
Life Sucks	72	123

Energy segue como variável mais significativa para o modelo e novamente um treino muito ruim. O modelo 2 passa a classificar ainda mais músicas como Life Sucks. Alone Again agora tem quase 71% de suas faixas classificadas erroneamente, enquanto que Life sucks chega a $86,01\%$ de acerto. Toda essa mudança favorecendo muito Life Sucks resulta apenas num pequeno aumento de erro. O modelo 2 tem erro de $37,56\%$.

- Treino - *random forest*

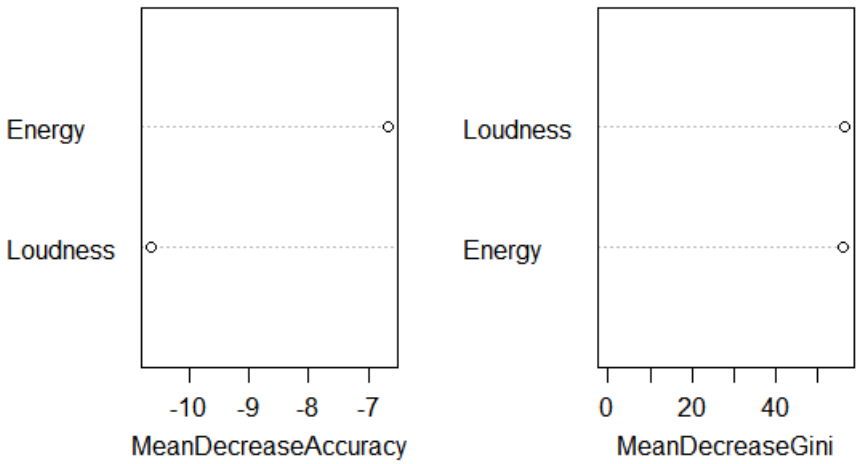

Figura 29: Importância de variáveis para o modelo 2.

Desta vez o *random forest* concorda com a regressão logística e *energy* também é tida como variável mais importante.

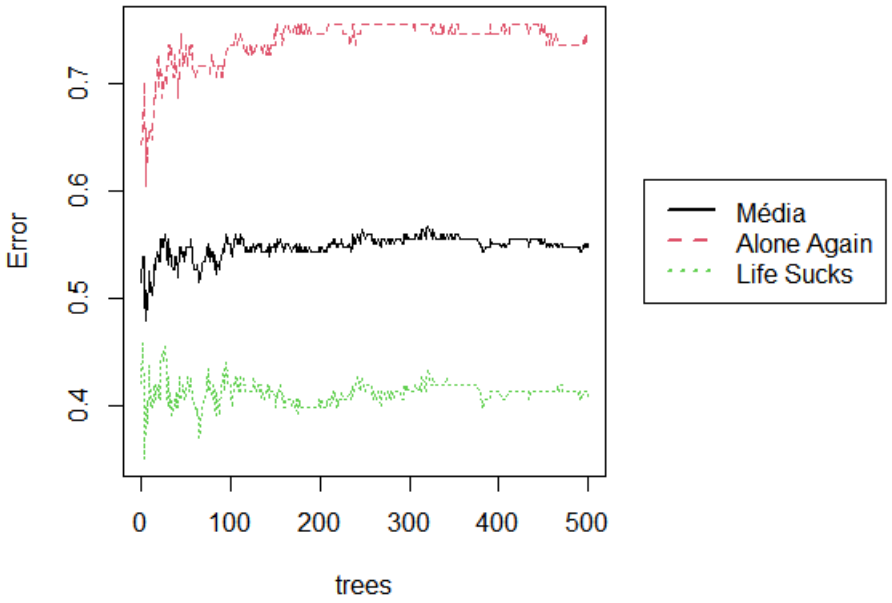

Figura 30: Erro do treino do *random forest* para o modelo 2.

Com poucas variáveis o modelo não demonstra muita evolução pelo aumento do número de árvores. Na verdade, com poucas árvores obteve-se um resultado melhor. E

perpetua-se a grande dificuldade para classificar corretamente as faixas de Alone Again.

Tabela 16: Matriz de confusão random forest do treino do modelo 2.

	Alone Again	Life Sucks
Alone Again	26	58
Life Sucks	76	85
class.error	0.75	0.41

O segundo método não aposta tanto em Life Sucks quanto o anteior, porém erra muito mais do que a regressão logística. Pelo resultado do treino o uso de apenas *energy* e *loudness* atrapalha bastante o método. Com erro de $54,69\%$ temos um modelo que erra mais do que acerta no seu treino.

- Previsões

Tabela 17: Previsão regressão logística modelo 2.

	Alone Again	Life Sucks
Alone Again	11	19
Life Sucks	37	38

Tabela 18: Previsão *random forest* modelo 2.

	Alone Again	Life Sucks
Alone Again	13	32
Life Sucks	35	25

Ambas as previsões são muito ruins novamente. Com a maior surpresa novamente ficando com o método de regressão. Após um treino de erro substancialmente menor do que o método de *random forest*, a Tabela 17 apresenta um erro de $53,33\%$. Possivelmente tivemos um sobreajuste do modelo aos dados de treino, mas já era esperando que tivéssemos um resultado fraco dada a similaridade das *playlists*. A previsão de *random forest* em 18 tem um resultado ainda pior que o seu treino e o resultado de seu rival. Com $62,86\%$ o método apresenta uma predisposição, assim como o anterior, de previsões ao acaso. Porém, ao errar quase 10% a mais que a regressão logística, *random forest* se torna mais interessante por se afastar da medida de 50%.

Ainda observamos uma grande dificuldade em treinar e prever para o par em

questão, mas *random forest* aparenta ter encontrado um caminho diferente para enxergar alguma diferença entre as *playlists* e obteve o melhor resultado.

4.4.3 Modelo 3 - Treino e previsão

Utilizando as variáveis *acousticness*, *energy*, *instrumentalness* e *loudness*.

- Treino - Regressão logística

Tabela 19: Matriz de confusão do treino do modelo 3.

	Alone Again	Life Sucks
Alone Again	40	23
Life Sucks	62	120

Novamente *energy* é tida como variável mais significativa para o modelo. Este treino tem um resultado que parece promissor. Seu erro de 34,69% parece baixo mas não passa confiança após as previsões ruins dos modelos anteriores.

- Treino - *random forest*

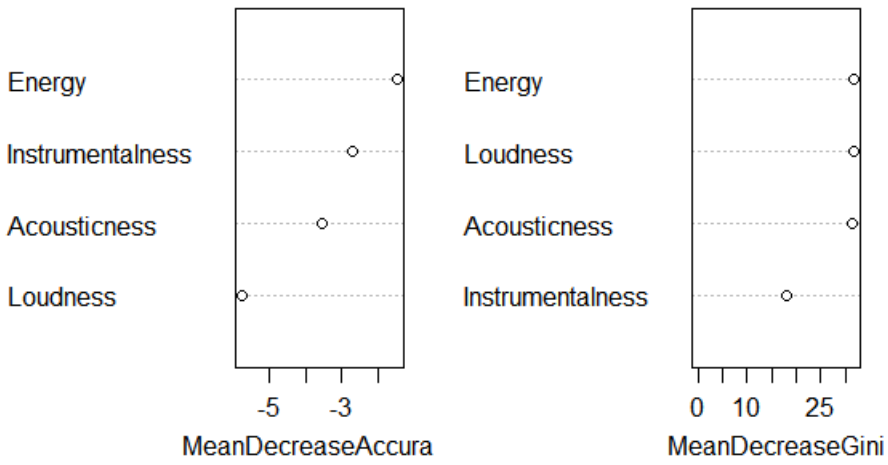

Figura 31: Importância de variáveis para o modelo 3.

Nos 2 últimos modelos ocorreu a concordância entre os métodos de que *energy* é a variável mais significativa para a análise. Recordando a Figura 4.1.2, poucas variáveis

mostram algum tipo de diferença entre as duas *playlists*. *Energy* é uma delas, mas ainda não parece, nem está sendo identificada como substancial para distinguir o par.

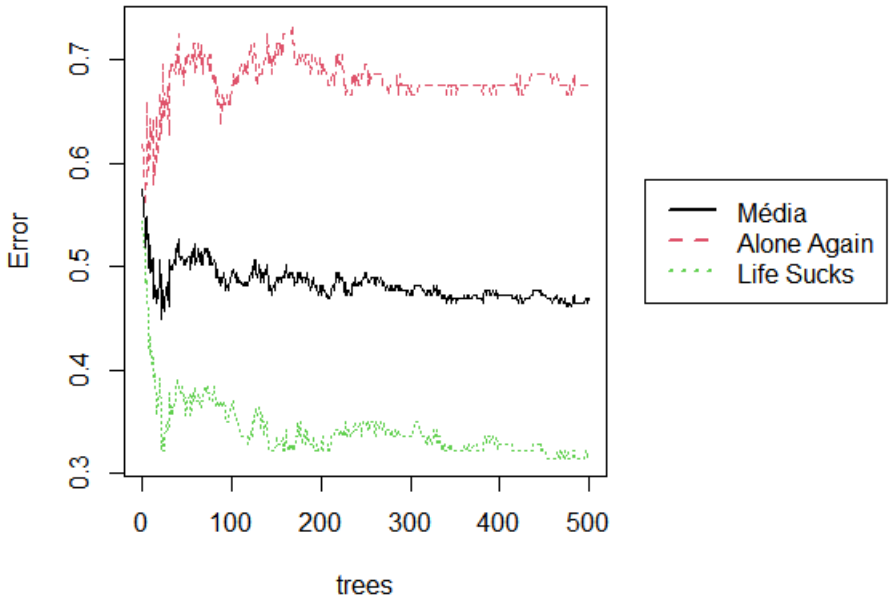

Figura 32: Erro do treino do *random forest* para o modelo 3.

Neste modelo, vê-se pela Figura 32 o erro com uma média que parece diminuir ao longo do incremento do número de árvores. Alone Again continua como o maior desafio de classificação e se estabiliza próximo após aproximadamente 300 árvores. A melhora da média de erro fica por parte de Life Sucks, indicando novamente uma maior facilidade para classificá-la.

Tabela 20: Matriz de confusão random forest do treino do modelo 3.

	Alone Again	Life Sucks
Alone Again	33	46
Life Sucks	69	97
class.error	0.68	0.32

Por fim, resultados novamente ruins e com a maioria das classificações para Life Sucks. Alone Again aparenta estar "escondida" ou contida dentro de Life Sucks segundo as variáveis utilizadas pelo trabalho. Com erro de $46,94\%$, como leve acalento, o treino ao menos acerta mais do que erra.

- Previsões

Tabela 21: Previsão regressão linear modelo 3.

	Alone Again	Life Sucks
Alone Again	14	12
Life Sucks	34	45

Tabela 22: Previsão *random forest* modelo 3.

	Alone Again	Life Sucks
Alone Again	14	26
Life Sucks	34	31

Não diferente das previsões anteriores, ambas são muito ruins. A regressão logística mais uma vez teve um treino que parecia promissor, mas, performou ainda muito mal no momento de previsão. Com erro de $43,81\%$, pelo menos voltou a acertar mais do que errar.

O *random forest* novamente errou mais do que acertou com $57,14\%$. Porém não errou tanto quanto no modelo 2, o que é, na verdade, uma piora de capacidade preditiva, visto que, quanto mais próximo de 50%, menos preditivo é o modelo.

No final, ambos tiveram previsões muito próximas, mas com o ajuste no resultado do *random forest*, é ele o método de melhor previsão para este modelo.

4.4.4 Comparação entre métodos - Alone Again vs Life Sucks

Deveras, após as análises, 3 modelos diferentes tiveram grande dificuldade em diferenciar as integrantes do segundo par. A descrição das *playlists* pelo Spotify indica climas e temas bem próximo. A análise inicial através dos boxplots na Figura 4.1.2 mostrou músicas de características mais similares ainda. Natural que os métodos tivessem grande dificuldade na distinção delas também.

Para um caso como esse que a existência de uma variável que fizesse referência ao conteúdo lírico da música seria, talvez, uma variável muito importante para ajudar na distinção das faixas. As características musicais descritas pelas variáveis nos modelos não são descriminantes o suficiente para as diferenciar.

Existe também a possibilidade de que essas *playlists* não sejam diferenciáveis, no

sentido de que músicas de Alone Again possam se encaixar em Life Sucks e vice-versa.

Além disso, interessante notar que para modelos de classificação o pior resultado possível é 50%, o que indica casualidade. Resultados próximos de 0%, por exemplo, indicam um modelo muito bom em fazer classificações opostas. Este tipo de erro pode facilmente ser corrigido com uma inversão dos valores atribuídos ás classes, assim transformando um modelo muito bom em errar num modelo muito bom em acertar.

Logo, tem-se o seguinte resultado: o método de regressão linear obteve os melhores treinos, porém não teve resultados bons na etapa de previsão; *random forest* teve as melhores previsões, todas com necessidade do ajuste de resultado.

4.5 Validação cruzada

- *K-folds*

Tabela 23: Melhores resultados validação cruzada por método.

CV	Método	Acurácia	Kappa
K-folds	Regressão logística	0.6000	0.1282
K-folds	*Random forest*	0.4800	-0.0860
LOO	Regressão logística	0.5886	0.1048
LOO	*Random forest*	0.4371	-0.1618

As validações cruzadas refletem muito bem quão difícil é distinguir esse par. Talvez as *playlists* realmente sejam tão similares por suas temáticas serem intercambiáveis entre elas. Ademais, *k-fold* e *LOO* concordam que o modelo 3 da regressão logística é quem performa melhor a distinção entre este par. Importante notar que o melhor Kappa dos métodos é do modelo 2 de *random forest*. Isso quer dizer que comparado aos outros modelos, este é que tem as previsões mais proporcionais entre as classes de interesse.

4.6 Modelagem Power Hour vs Spooning

Para o último par, assim como no primeiro, o modelo utilizando todas as variáveis não converge. Dessa vez *energy* é a variável que separa os dados quase completamente e será retirada do estudo. Nesta análise, as faixas são classificadas entre Power Hour (0) e Spooning (1).

4.6.1 Modelo 1 - Treino e previsão

Utilizando as variáveis *acousticness*, *danceability*, *duration*, *instrumentalness*, *key*, *liveness*, *loudness*, *mode*, *speechiness*, *tempo* e *valence*.

- Treino - Regressão logística

Dentro das variáveis compreendidas no modelo, *acousticness* e *loudness* são indicadas como as mais, mas não únicas, variáveis significativas. Spooning por ter uma proposta mais intimista, até romântica, tem uma maior preferencia por faixas acústicas e de menor volume. Power Hour é um *playlist* focada na prática de exercícios e propõe músicas mais barulhentas e pouco ou nada acústicas. Faz sentido essas variáveis serem as mais significativas.

Tabela 24: Matriz de confusão regressão logística do modelo 1.

	Power Hour	Spooning
Power Hour	74	4
Spooning	4	98

Seu treino tem resultado muito bom com pequeno erro de $4,44\%$. Quase perfeito graças a uma distinção clara entre o par.

- Treino - *random forest*

Pelo segundo método, as mesmas duas variáveis como as mais importantes e *key* e *mode* como menos importantes.

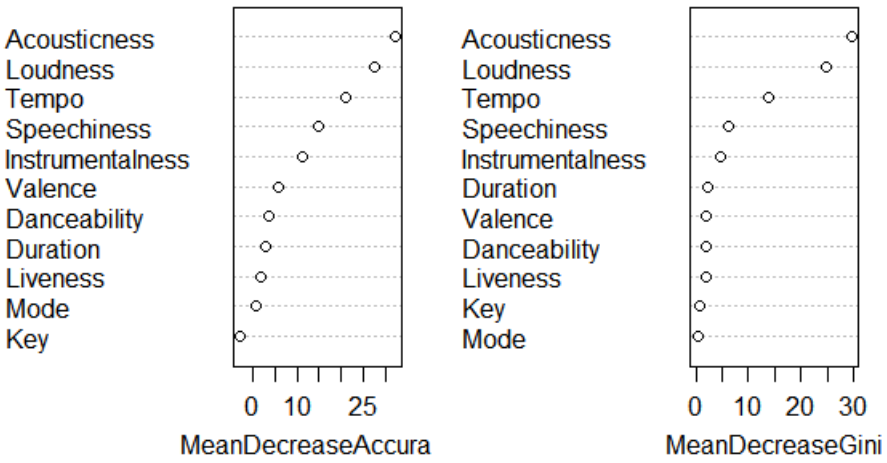

Figura 33: Importância de variáveis para o modelo 1.

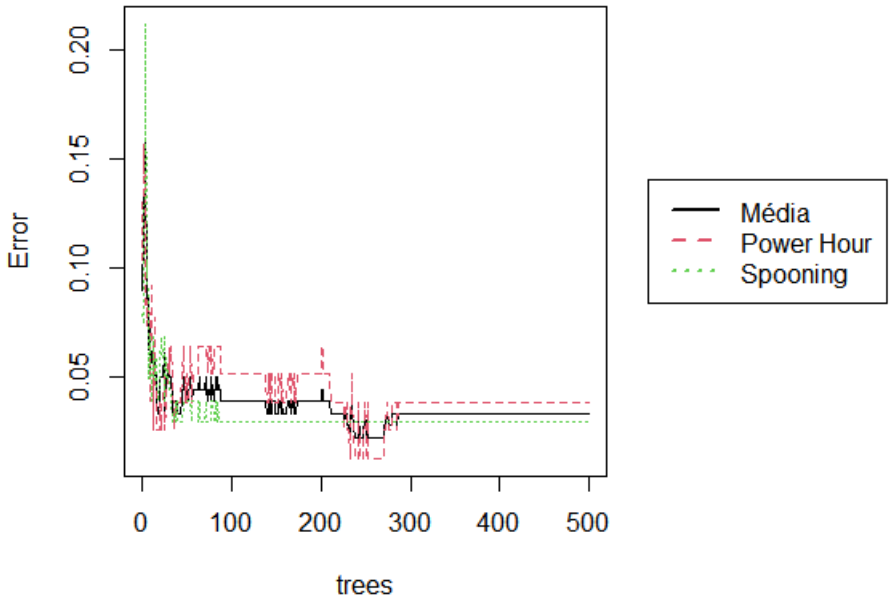

Figura 34: Erro do treino do *random forest* para o modelo 1.

A maior facilidade na distinção das *playlists* também é visualizada pelo gráfico do erro de classificação na Figura 34. Pela primeira vez o estudo mostra uma alteração, mesmo que momentânea, de qual *playlist* possui o menor erro no treino. De maneira geral ambas possuem erros muito próximos e baixos.

O método de *random forest* também tem treino de resultado muito bom. Erro baixo de apenas $2,78\%$. Mais baixo que o método anterior.

Tabela 25: Matriz de confusão random forest do treino do modelo 1.

	Power Hour	Spooning
Power Hour	75	3
Spooning	3	99
class.error	0.04	0.03

- Previsões

Tabela 26: Previsão regressão logística modelo 1.

	Power Hour	Spooning
Power Hour	21	4
Spooning	1	44

Tabela 27: Previsão *random forest* do modelo 1.

	Power Hour	Spooning
Power Hour	21	1
Spooning	1	47

Como esperado após dois treinos muito bons, ambos os métodos fizeram previsões muito boas. *Random forest* teve o melhor resultado com apenas $2,86\%$ de erro, enquanto que seu rival teve $7,14\%$, mais do que o dobro. As *playlists* aparentam ser muito bem diferenciáveis com as características descritas pelo banco de dados.

4.6.2 Modelo 2 - Treino e previsão

Utilizando *acousticness*, *duration*, *instrumentalness*, *liveness* e *loudness*.

- Treino - Regressão logística

Acousticness é a variável mais significativa, também seguida por *loudness*.

Treinamento tem resultado quase perfeito novamente com erro de 4,44%. Curiosamente igual ao modelo anterior, mas com uma melhora para a Power Hour e piora para Spooning em 1 faixa cada.

Tabela 28: Matriz de confusão do treino do modelo 2.

	Power Hour	Spooning
Power Hour	75	5
Spooning	3	97

- Treino - *random forest*

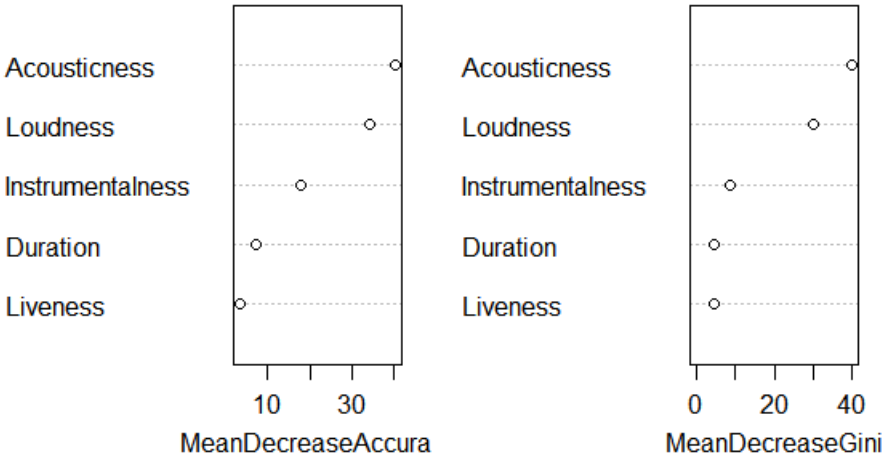

Figura 35: Importância de variáveis para o modelo 2.

Mais uma vez há a concordância entre os modelos sobre as variáveis mais significantes: *acousticness* e *loudness*.

Pela segunda vez a alternância entre qual classe é mais fácil e mais difícil de ser prevista. Entre 100 e 200 árvores o modelo atinge uma estabilidade de erro com Power Hour como a *playlist* de menor erro, situação contrária ao modelo anterior.

Tabela 29: Matriz de confusão random forest do treino do modelo 2.

	Power Hour	Spooning
Power Hour	73	8
Spooning	5	94
class.error	0.06	0.08

Assim como no método de regressão linear, tem-se um treino muito bom. O erro de $7,22\%$ é quase o dobro do método anterior, porém ainda é bem baixo.

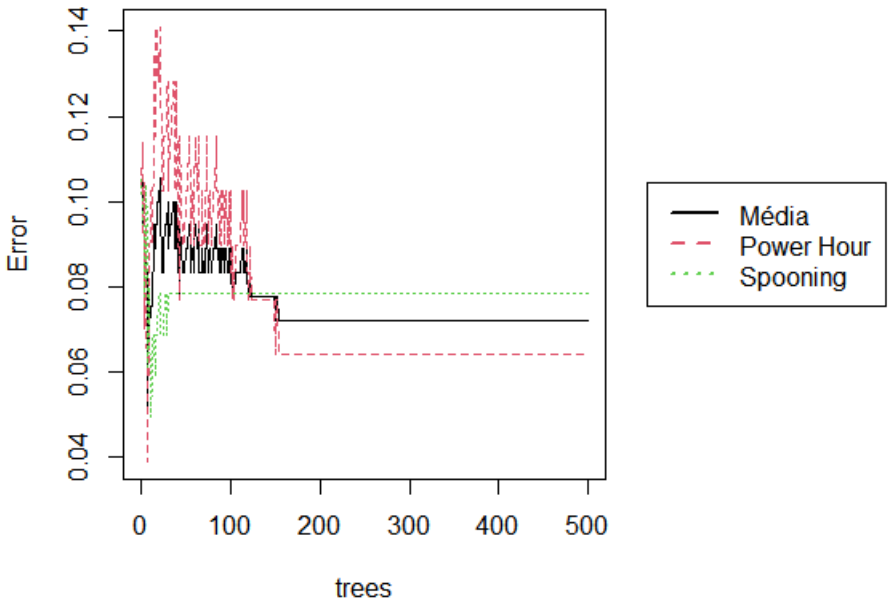

Figura 36: Erro do treino do *random forest* para o modelo 2.

- Previsões

Tabela 30: Previsão regressão logística modelo 2.

	Power Hour	Spooning
Power Hour	21	3
Spooning	1	45

Tabela 31: Matriz de confusão random forest da previsão do modelo 2.

	Power Hour	Spooning
Power Hour	21	4
Spooning	1	44

As previsões de ambos os métodos vão muito bem. Com ambos os erros abaixo de 10% percebe-se que as variáveis presentes nos modelos também representam muito bom bem as diferentes características de cada *playlist*. Regressão logística possui o melhor resultado com apenas $5,71\%$ de erro, contra $7,14\%$ de *random forest*.

4.6.3 Comparação entre métodos - Power Hour vs Spooning

Indo de acordo com a descrição de cada *playlists* e a análise descritiva dos boxplots das suas variáveis, ambos os métodos de classificação obtiveram êxito na distinção entre as suas faixas. Tanto nos treinos quanto na previsões se obteve erros inferiores a 10%, evidenciando o desempenho de ambos os métodos como muito consistentes. A diferença entre os objetivos das *playlists* é tranquilamente enxergada por ambos os modelos dos métodos em questão.

O método com maior facilidade e melhor resultado foi *random forest* com seu primeiro modelo, mas seguido bem de perto pela regressão.

4.7 Validação cruzada

Tabela 32: Melhores resultados validação cruzada por método.

CV	Método	Acurácia	Kappa
K-folds	Regressão logística	0.9320	0.8595
K-folds	*Random forest*	0.9640	0.9240
LOO	Regressão logística	0.9400	0.8756
LOO	*Random forest*	0.9640	0.9249

As validações cruzadas convergem para a mesma conclusão: o melhor método para o último par é *random forest* com o modelo 1 utilizando 2 variáveis disponíveis nos nós. Como este par é bem distinto, até o pior modelo teve resultado muito bom.

4.8 Modelagem todas as *playlists*

Por fim, o maior desafio para os métodos de classificação ao se trabalhar com todas as *playlists* ao mesmo tempo. Para a regressão logística ainda foi utilizada a família binomial, então, testou-se a capacidade do método de separar 1 *playlists* de cada vez do conjunto de *playlists*. *Random forest* por sua vez, já foi mais direto e trabalhou classificando todas as *playlists* ao mesmo tempo. Então, ocorrreu o teste de alguns modelos para cada *playlist* na regressão logística e um único modelo no *random forest*. Como já aconteceu uma análise mais detalhada no estudo dos 3 pares, aqui tem-se uma visão mais abreviada dos resultados.

O primeiro método de classificação utilizado é o de regressão logística, assim como nos pares passados. Como já explicado, dessa vez é analisada 1 *playlist* contra todas as outras, e repetir até que todas as *playlists* sejam comparadas contra o resto. Então, a classificação será: *Playlist* de interesse (0) e Outras (1).

Seguindo pela ordem alfabética, o início é com Alone Again (0) e Outras (1), até o término em Spooning (0) e Outras (1).

4.8.1 Alone Again vs Outras - Regressão logística

- Treino

Tabela 33: Treino para Alone Again vs Outras.

	Alone Again	Beast Mode	Life Sucks	Piano Relaxante	Power Hour	Spooning
Alone Again	15	0	14	0	0	4
Outras	90	141	126	248	72	108

Pelo treino, resultados interessantes. Relembrando a análise dos boxplots na Secção 4.1.4 Alone Again, Life Sucks e Spooning davam indícios de pertencerem a um mesmo grupo de *playlists* com características muito semelhantes descritas por suas variáveis. O treino mostra mostra grande dificuldade para diferenciar Alone Again de Life Sucks, semelhante ao estudo apenas entre os pares, e agora também tem dificuldade, em menor grau, com Spooning.

- Previsão

Tabela 34: Previsão para Alone Again vs Outras.

	Alone Again	Beast Mode	Life Sucks	Piano Relaxante	Power Hour	Spooning
Alone Again	9	0	9	0	0	1
Outras	36	59	51	99	28	37

Para o momento de previsão, vê-se um resultado que não foge do treino, com as observações erroneamente classificadas como Alone Again pertencerem a Life Sucks e Spooning.

Ao testar a capacidade do modelo de diferenciar a *playlist* Alone Again de todas as outras, é necessário tomar muito cuidado com as medidas de acerto e erro gerais. O

Tabela 35: Previsão simplificada para Alone Again vs Outras.

	Alone Again	Outras
Alone Again	9	10
Outras	36	274

desbalanço entre a quantidade de observações das duas classes acaba distorcendo tais medidas retornando bons números. Um erro de apenas $13,98\%$ à primeira vista parece o resultado de um bom modelo. Porém, ao se conferir a medida de especificidade(verdadeiro negativo) e sensibilidade(verdadeiro positivo), constata-se uma taxa de 20% de verdadeiro positivo e mais de 95% de verdadeiro negativo em ambos os modelos. Esses valores mostram como o grande número de acertos na categoria "Outras" afeta as medidas mais simples de acurácia.

4.8.2 Beast Mode vs Outras - Regressão logística

- Treino

Tabela 36: Treino para Beast Mode vs Outras.

	Beast Mode	Alone Again	Life Sucks	Piano Relaxante	Power Hour	Spooning
Beast Mode	103	1	2	0	36	1
Outras	38	104	138	248	36	111

Além da esperada confusão entre Beast Mode e Power Hour, algumas faixas de Alone Again, Life Sucks e Spooning também são classificadas pelos modelos na fase de treino como Beast Mode, mesmo que em menor grau.

- Previsão

Tabela 37: Previsão para Beast Mode vs Outras.

	Beast Mode	Alone Again	Life Sucks	Piano Relaxante	Power Hour	Spooning
Beast Mode	43	0	1	0	15	1
Outras	16	45	59	99	13	37

Os resultados são bem melhores que a *playlist* anterior. Erro geral de apenas $10,03\%$ no modelo 1 e $10,64\%$ no 2. Apesar do grande desbalanço entre os grupos a taxa

Tabela 38: Previsão simplificada para Beast Mode vs Outras.

	Beast Mode	Outras
Beast Mode	43	17
Outras	16	253

de verdadeiro positivo é 72,8% no modelo 1 e 69, 49% no 2. E como esperado, a maior parte dos erros da previsão acontecem com Power Hour, a outra *playlist* focada em prática de exercícios.

4.8.3 Life Sucks vs Outras - Regressão logística

- Treino

Tabela 39: Treino para Life Sucks vs Outras.

	Life Sucks	Alone Again	Beast Mode	Piano Relaxante	Power Hour	Spooning
Life Sucks	24	24	0	0	0	8
Outras	116	81	141	248	72	104

Treinos dos modelos mostram agora, pelo ponto de vista de Life Sucks, a sua fortíssima similaridade com Alone Again e em grau menor com Spooning. Isso só reforça a percepção de que as 3 *playlists* formam um grupo de similaridade muito alta.

- Previsão

Tabela 40: Previsão para Life Sucks vs Outras.

	Life Sucks	Alone Again	Beast Mode	Piano Relaxante	Power Hour	Spooning
Life Sucks	11	15	0	0	0	3
Outras	49	30	59	99	28	35

Tabela 41: Previsão simplificada para Life Sucks vs Outras.

	Life Sucks	Outras
Life Sucks	11	18
Outras	49	251

Os modelos se comportam mal para classificar corretamente as faixas como Life Sucks se confundindo muito com Alone Again e um pouco com Spooning. Curiosamente, se saem melhor do que os modelos de Alone Again para classificar as músicas como Alone Again.

4.8.4 Piano Relaxante vs Outras - Regressão logística

- Treino

Tabela 42: Treino para Piano Relaxante vs Outras.

	Piano Relaxante	Alone Again	Beast Mode	Life Sucks	Power Hour	Spooning
Piano Relaxante	247	0	0	1	0	1
Outras	1	105	141	139	72	111

Semelhante ao momento de análise dos pares, Piano Relaxante continua se destacando muito facilmente das *playlists* restantes. Alone Again, Life Sucks e Spooning são as *playlists* que causam alguma mínima confusão aos modelos. Pelas propostas mais calma dessas *playlists*, faz sentido isso ocorrer, mesmo que num nível tão pequeno.

- Previsão

Tabela 43: Previsão para Piano Relaxante vs Outras.

	Piano Relaxante	Alone Again	Beast Mode	Life Sucks	Power Hour	Spooning
Piano Relaxante	99	1	0	1	0	0
Outras	0	44	59	59	28	38

Tabela 44: Previsão simplificada para Piano Relaxante vs Outras.

	Piano Relaxante	Outras
Piano Relaxante	99	2
Outras	0	228

Assim como no seu treino, os modelos separam muito bem Piano Relaxante do resto, com uma pequena quantidade de erros classificando músicas de Alone Again e Life Sucks como *playlist* de interesse. É a *playlist* que tem melhor resultado. Erro de apenas $1,22\%$ com precisão e sensibilidade acima de 97%.

4.8.5 Power Hour vs Outras - Regressão logística

- Treino

Tabela 45: Treino do modelo 2 para Power Hour vs Outras.

	Power Hour	Alone Again	Beast Mode	Life Sucks	Piano Relaxante	Spooning
Power Hour	36	0	18	0	0	0
Outras	36	105	123	140	248	112

A segunda *playlist* focada em atividade física tem resultado similar à primeira. Diferente de Beast Mode, os modelos só a confundem com a própria Beast Mode.

- Previsão

Tabela 46: Previsão do modelo 2 para Power Hour vs Outras.

	Power Hour	Alone Again	Beast Mode	Life Sucks	Piano Relaxante	Spooning
Power Hour	15	0	9	0	0	0
Outras	13	45	50	60	99	38

Tabela 47: Previsão do modelo 2 simplificada para Power Hour vs Outras.

	Power Hour	Outras
Power Hour	15	9
Outras	13	292

Assim como previsto pela análise dos boxplots, Power Hour se confunde com Beast Mode.

Os modelos tem certo grau de qualidade para classificação. Não performam tão bem quanto os modelo de Beast Mode, mas os 2 melhores modelos têm taxa de verdadeiro positivo próximo de 50% e precisão próxima de 63%.

4.8.6 Spooning vs Outras - Regressão logística

- Treino

Tabela 48: Treino do modelo 1 para Spooning vs Outras.

	Spooning	Alone Again	Beast Mode	Life Sucks	Piano Relaxante	Power Hour
Spooning	4	2	0	6	0	0
Outras	108	103	141	134	248	72

De longe o pior treino do estudo. Pequena quantidade de faixas classificadas como a classe de interesse. As classificações incorretas que ocorreram foram entre Alone Again e Life Sucks.

- Previsão

Tabela 49: Previsão do modelo 1 para Spooning vs Outras.

	Spooning	Alone Again	Beast Mode	Life Sucks	Piano Relaxante	Power Hour
Outras	38	45	59	60	99	28

Pela primeira vez o modelo não classificou qualquer música como a *playlist* de interessante. Apesar disso, graças ao desbalanço da quantidade de observações entre os possíveis eventos do estudo, obteve-se um erro baixo de apenas $11,55\%$. Spooning está completamente contida entre as outras *playlists*. A descrição de suas músicas por parte das variáveis do estudos não possui qualquer particularidade que a diferencie das outras.

4.9 Modelagem todas as *playlists* - *random forest*

4.9.1 Treino e ajuste do modelo - *random forest*

Tabela 50: Matriz de confusão *random forest* do treino do modelo 1

	Alone Again	Beast Mode	Life Sucks	Piano Relaxante	Power Hour	Spooning
Alone Again	20	0	43	0	0	16
Beast Mode	2	114	10	0	42	8
Life Sucks	66	3	55	0	0	38
Piano Relaxante	0	0	0	248	0	2
Power Hour	0	22	0	0	30	0
Spooning	17	2	32	0	0	48
class.error	0.81	0.19	0.61	0.00	0.58	0.57

Na etapa de treino já se vê uma interessante diferença entre os métodos. Pelo *random forest* a *playlist* Spooning consegue ser bem identificada. Uma semelhança interessante é a dificuldade de separar Alone Again, Life Sucks e Spooning, Beast Mode e Power Hour. Piano Relaxante se destaca muito bem em ambos os métodos.

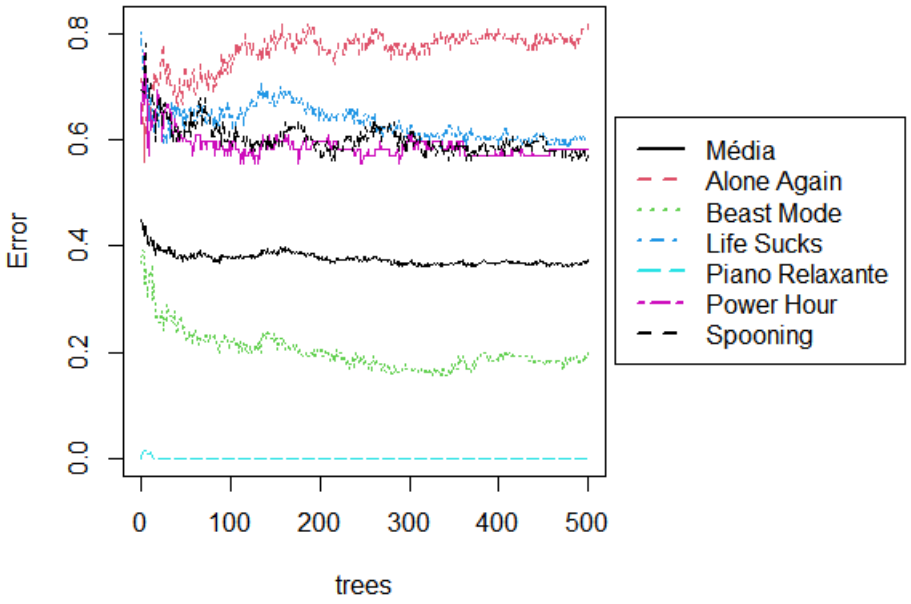

Figura 37: Erro do treino do *random forest* para o modelo 1.

Analisando o gráfico na Figura 37, o modelo tem um erro médio bem estável, Piano Relaxante está sempre bem distante das outras *playlists* e junto com Beast Mode são as únicas *playlists* a ter erro de classificação inferior a 50%. Apesar de Alone Again, Life Sucks e Spooning se confundirem bastante, Alone Again se distancia com o pior erro, e as outras 2 estão numa faixa de erro muito próxima de Power Hour que só se confunde com Beast Mode.

- Previsões

Tabela 51: Previsão *random forest*

	Alone Again	Beast Mode	Life Sucks	Piano Relaxante	Power Hour	Spooning
Alone Again	9	0	23	0	0	9
Beast Mode	4	43	7	0	18	3
Life Sucks	27	4	22	0	0	13
Piano Relaxante	0	0	0	99	0	0
Power Hour	0	9	0	0	10	0
Spooning	5	3	8	0	0	13

Alone Again teve o pior desempenho (sensibilidade de 20%, precisão de $21,95\%$). Poucos acertos e o modelo parece estar mais refinado para acertar a classificação de Life Sucks.

Beast mode tem um desempenho bom (sensibilidade de $72,88\%$, precisão de $57,33\%$) e seu erro mais substancial é com Power Hour, como esperado.

Life Sucks tem um desempenho mediano para fraco(sensibilidade de $36,67\%$, precisão de $33,33\%$), mas se confunde muito com Alone Again e em grau menor com Spooning. Inclusive prevê melhor para Alone Again. Seus previsão seriam melhor se seus modelos de tomada de decisão fossem invertidos.

Piano Relaxante tem desempenho perfeito graças a ser muito diferente das outras *playlists*.

Power Hour tem um desempenho mediano para fraco (sensibilidade de $35,71\%$, precisão de $52,63\%$) pois se confunde muito com Beast Mode. Curiosamente Power Hour só se confunde com Beast Mode, mas Beast Mode se confunde com outras *playlists*. Pode-se entender que todas as faixas de Power Hour poderiam fazer parte de Beast Mode, mas o contrário não é verdade.

Por este método Spooning tem um desempenho bem superior ao passado, porém ainda é um resultado de mediano para fraco (sensibilidade de $34,21\%$, precisão de $44,83\%$). Se confunde muito com Life Sucks e em menor grau com Alone Again e Beast Mode.

Mais uma vez se constata que a maior dificuldade dos métodos está em separar Beast Mode de Power Hour e Alone Again de Life Sucks e de Spooning. Piano Relaxante se separa muito bem de todas as outras.

O método *random forest* tem $59,57\%$ de acurácia em sua previsão. Um resultado muito próximo do acaso e mostra a complexidade de distinguir *playlists*, sendo algumas de características tão similares, com as variáveis disponíveis no momento.

5 Conclusão

O objetivo deste trabalho foi entender o funcionamento, os pros e contras de algoritmos de classificação aplicados a um situação de dia-a-dia: escolher a música certa para o momento certo.

De maneira geral, os métodos de regressão logística e $randomforest$ obtiveram resultados muito interessantes, tanto no treinamento quanto nas previsões e cada um teve seu momento como melhor e como pior método.

Uma vantagem clara para a regressão logística é sua interpretabilidade e seu tempo de execução, principalmente quando combinada com a validação cruzada *leave one out*. *Random forest* tem um custo computacional maior sem trazer o equivalente em acurácia de resultados e menos interpretável, porém, foi o único método que conseguiu identificar Spooning em meios às outras playlists.

Os métodos de validação cruzada trouxeram uma robustez importante mostrando a influência que uma escolha viesada pode ter no resultado das metodologias de classificação utilizadas neste trabalho. Também revelaram uma situação mais fidedigna para o contexto de seleção de músicas para proposta específica.

Quanto aos resultados entre os pares, os métodos performaram muito bem quando aplicados aos pares 1 e 3, porém tiveram muita dificuldade com o par 2.

Analisando as variáveis disponíveis para o trabalho e as descrições das variáveis, possivelmente a adição de uma variável como "tema lírico" ajudaria a melhorar os resultados do segundo par e . Entretanto, é possível que 2 *playlists* tenham temas em comum com propostas musicais semelhantes e essencial suas músicas sejam intercambiáveis, tornando a distinção entre elas impossível ou incorreta.

Outra variável que poderia ser interessante ao trabalho é o gênero da música. Até o momento, o Spotify só informa os gênero dos artistas participantes da faixa o que não é muito interessante quando analisando as músicas isoladamente. Além disso, outros fatores que poderiam ser determinantes na categorização de faixas são período e localização de lançamento.

Já na análise conjunta das 6 *playlists*, apesar dos erros de classificação intragrupais no grupo 1 (Alone Again, Life Sucks e Spooning) e 2 (Beast Mode e Power Hour), poucas vezes ocorreu a confusão entre os grupos. Sendo o melhor exemplo o grupo 3 (Piano Relaxante) constatando-se a capacidade dos métodos em distinguir bem os grandes

grupos.

Para o futuro, é importante a exploração de outros algoritmos de classificação como redes neurais, *support vector machines* e análise de discriminante linear, uma abrangência maior de *playlists* e o teste dos modelos com faixas alheias às seleções em estudo.

Referências

BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001.

CORMEN, T. H. et al. *Introduction to algorithms*. [S.l.]: MIT press, 2009.

IZBICKI, R.; SANTOS, T. M. dos. *Aprendizado de máquina: uma abordagem estatística*. [S.l.: s.n.], 2020. ISBN 978-65-00-02410-4.

KASSAMBARA, A. *Machine learning essentials: practical guide in R: CreateSpace independent publishing platform*. 2018.

MIGUEL, T. *Arvore de Decisão em R.* 2020. Disponível em: ¡https://aprenderdatascience.com/regressao-logistica/¿. Acessado em 13 set. 2021.

RIPLEY, B. D. *Pattern recognition and neural networks*. [S.l.]: Cambridge university press, 2007.

SPOTIFY. *Spotify for developers*. Disponível em: <https://developer.spotify.com/documentation/web-api/>. Acessado em 28 ago. 2021.

SPOTIFY. *Spotify for developers*. Disponível em: <https://developer.spotify.com/documentation/web-api/reference/#objects-index>. Acessado em 28 ago. 2021.

SPOTIFY. *Tipos de playlists do Spotify*. 2019. Disponível em: <https://artists.spotify.com/pt/help/article/types-of-spotify-playlists.>. Acessado em 30 ago. 2021.

STANKEVIX, G. *Arvore de Decisão em R.* 2019. Disponível em: ¡https://medium.com/@gabriel.stankevix/arvore-de-decis%C3%A3o-em-r-85a449b296b2¿. Acessado em 16 set. 2021.

TEAM, R. C. et al. R: A language and environment for statistical computing. Vienna, Austria, 2013.

WIDMANN, M. *Cohen's Kappa: Learn It, Use It, Judge It.* 2020. Disponível em: <https://www.knime.com/blog/cohens-kappa-an-overview#: :text=Cohen%27s%20kappa%20is%20a%20metric,performance%20of%20a%20classification%20model.>. Acessado em 13 jul. 2022.

WITTEN, I. H. et al. Practical machine learning tools and techniques. In: *DATA MINING*. [S.l.: s.n.], 2005. v. 2, p. 4.