HARMONIZANDO COM O PÚBLICO: UM ESTUDO DAS VARIÁVEIS QUE IMPACTAM A POPULARIDADE DE MÚSICAS NO SPOTIFY

ANTONIO PATRICIO QUERES NETO
Instituto Federal do Espírito Santo. E-mail: antoniopatricioqn@gmail.com

GUILHERME GUILHERMINO NETO
Instituto Federal do Espírito Santo. E-mail: guilherme.neto@ifes.edu.br

1. PROBLEMA DE PESQUISA

O *streaming* de música revolucionou a forma como as pessoas consomem música, permitindo o acesso a um vasto catálogo sob demanda, mediante pagamento de assinatura ou em plataformas gratuitas com anúncios. O número de reproduções de músicas em aplicativos de *streaming*, conhecido como *streams*, tornou-se um indicador crucial de popularidade e sucesso no mercado musical.

O mercado fonográfico global tem experimentado um crescimento expressivo nos últimos anos, impulsionado principalmente pelo aumento das receitas de *streaming*. Em 2023, o mercado global de assinaturas de música atingiu a marca de 713,4 milhões de assinantes, um aumento significativo em relação aos anos anteriores, com o Spotify liderando o segmento, representando 31,7% do mercado e ultrapassando a marca de 226 milhões de assinantes (Mulligan, 2024). No Brasil, o mercado fonográfico atingiu R$2,5 bilhões em 2022, um aumento de 15,4% em relação ao ano anterior, sendo 86,2% desse valor proveniente do *streaming* (Franco, 2022).

Esse crescimento exponencial do *streaming* de música torna o cenário ainda mais competitivo para artistas e produtores musicais. O sucesso nesse ambiente não depende apenas da qualidade da produção, mas também da capacidade de compreender e atender às preferências do público. Diante disso, este estudo busca identificar as variáveis (características) que influenciam a popularidade das músicas no Spotify, a fim de fornecer *insights* valiosos para a indústria musical.

A pergunta de pesquisa que norteará este estudo é: **Quais características das músicas influenciam significativamente o número de *streams* no Spotify?**

Este trabalho é relevante por oferecer uma abordagem inovadora baseada em dados para a criação musical, capacitando os artistas a tomar decisões mais estratégicas e aumentar suas chances de sucesso no *streaming*. Ao compreender

as preferências do público a partir dos dados, os músicos podem personalizar suas criações para obter maior popularidade.

O objetivo principal deste estudo é desenvolver um modelo de regressão múltipla que possa identificar quais são as variáveis que têm maior influência na popularidade de músicas no Spotify e por consequência no volume de reproduções (*streams*). Para compreender melhor a estrutura e as relações presentes na base de dados, realizamos uma análise exploratória onde analisou-se o histórico de reprodução, listas de reprodução favoritas e interações dos usuários no Spotify para compreender as variáveis e as tendências nas preferências dos usuários.

Espera-se que este modelo forneça *insights* acionáveis para músicos e produtores musicais, auxiliando-os na criação de conteúdo musical mais relevante e atrativo para o público.

2. PROCESSOS METODOLÓGICOS

A metodologia adotada nesta pesquisa baseia-se em considerações teóricas e práticas sobre a Análise de Regressão Múltipla e seguirá uma abordagem sistemática baseada no processo de construção de modelo conforme descrito por Hair Jr. et al. (2009).

Outro caminho metodológico possível seria uma análise da correlação de *Pearson* e *Spearman*, que são úteis para medir a força e a direção da associação linear entre duas variáveis, no entanto, elas não permitem quantificar o efeito de cada variável independente sobre a variável dependente, nem construir um modelo preditivo com múltiplas variáveis. A escolha do método de Análise de Regressão Múltipla, portanto, se justifica pela sua capacidade de identificar a existência de relações entre as variáveis, e também estimar a magnitude e a direção dessas relações, o que é essencial para responder à pergunta de pesquisa.

Este estudo não envolve a coleta de dados sensíveis e respeitará os princípios éticos de pesquisa. A base de dados utilizada, disponibilizada por Elgiriyewithana (2023) na plataforma Kaggle, contém dados anonimizados sobre as músicas mais reproduzidas no Spotify em 2023, garantindo a privacidade dos usuários da plataforma.

A tabela abaixo apresenta as variáveis presentes na base de dados bem como informações adicionais de descrição e sobre o formato original dos dados.

INSTITUTO FEDERAL
Espírito Santo
Campus Cariacica

Variáveis da Base de Dados

Nome da Variável	Descrição da Variável	Tipo da Variável	Tipo do Dado	Valores Únicos
streams	Número total de streams no Spotify	dependente	object	949
track_name	Nome da música	independente	object	943
artist(s)_name	Nome do(s) artista(s) da música	independente	object	645
artist_count	Número de artistas que contribuíram para a música	independente	int64	8
released_year	Ano em que a música foi lançada	independente	int64	50
released_month	Mês em que a música foi lançada	independente	int64	12
released_day	Dia do mês em que a música foi lançada	independente	int64	31
in_spotify_playlists	Número de playlists do Spotify nas quais a música está incluída	independente	int64	879
in_spotify_charts	Presença e classificação da música nas paradas do Spotify	independente	int64	82
in_apple_playlists	Número de playlists do Apple Music nas quais a música está incluída	independente	int64	234
in_apple_charts	Presença e classificação da música nas paradas musicais da Apple	independente	int64	172
in_deezer_playlists	Número de playlists do Deezer em que a música está incluída	independente	object	348
in_deezer_charts	Presença e posição da música nas paradas da Deezer	independente	int64	34
in_shazam_charts	Presença e classificação da música nas paradas do Shazam	independente	object	198
bpm	Batidas por minuto, uma medida do andamento	independente	int64	124
key	Tom da música	independente	object	11
mode	Modo (maior ou menor)	independente	object	2
danceability_%	Porcentagem que indica quão adequada a música é para dançar	independente	int64	72
valence_%	Positividade do conteúdo musical	independente	int64	94
energy_%	Nível de energia percebido	independente	int64	80
acousticness_%	quantidade de som acústico	independente	int64	98
instrumentalness_%	Quantidade de conteúdo instrumental	independente	int64	39
liveness_%	Presença de elementos de performance ao vivo	independente	int64	68
speechiness_%	Quantidade de palavras faladas	independente	int64	48

Para realizar esta pesquisa, foram utilizadas diversas ferramentas e recursos, incluindo a linguagem de programação Python e SQL, além de ferramentas como PySpark, Synapse Data Science e Google Colab. Essas ferramentas oferecem a capacidade de realizar análises avançadas de dados e construir modelos de regressão múltipla de forma eficiente.

2.1. Análise Exploratória da Base de Dados

Para compreender melhor a estrutura e as relações presentes na base de dados, realizamos uma análise exploratória inicial. Essa etapa foi crucial para obter *insights* sobre as características dos dados e identificar padrões relevantes para o modelo de regressão.

Primeiramente, realizamos uma análise univariada das variáveis numéricas, o que nos permitiu visualizar a distribuição de cada variável individualmente. Para isso, utilizamos histogramas e gráficos de densidade, que nos forneceram informações sobre a forma, a centralidade e a dispersão dos dados, conforme se pode verificar nas figuras 1 e 2 do apêndice.

Para identificar *outliers*, empregamos dois métodos: o método do Intervalo Interquartil (IQR) e o Z-score. No método IQR, calculamos o primeiro quartil (Q1) e o terceiro quartil (Q3) de cada variável. Definimos os limites inferior e superior como Q1 - 1.5IQR e Q3 + 1.5IQR, respectivamente. Valores fora desses limites foram considerados *outliers*. No método Z-score, calculamos a média e o desvio padrão de cada variável, identificando como *outliers* os valores que apresentaram Z-score superior a 3 ou inferior a -3. Esses procedimentos foram cruciais para detectar e remover os *outliers* do modelo e garantir que o desempenho do modelo de regressão não fosse prejudicado.

A Matriz de Correlação das Variáveis Numéricas nos permitiu analisar a força e a direção da relação linear entre pares de variáveis numéricas. Os valores próximos de 1 indicam uma forte correlação positiva, valores próximos de -1 indicam uma forte correlação negativa e valores próximos de 0 indicam uma correlação fraca ou inexistente.

2.2. Planejamento da Análise de Regressão

O planejamento da análise de regressão envolveu a seleção das variáveis a serem incluídas no modelo, a identificação e tratamento de *outliers* e multicolinearidade, e a transformação de variáveis para garantir a adequação aos pressupostos do modelo de regressão linear múltipla.

Após a estimação do modelo de regressão em sua primeira versão, foi crucial avaliar a significância estatística de cada variável independente. Essa etapa ajudou a identificar quais variáveis têm um impacto estatisticamente relevante no número de *streams* e quais não contribuem significativamente para o modelo. Para isso, o código verifica os valores-p (P>|t|) associados a cada variável, excluindo o valor constante (*const*). O valor-p representa a probabilidade de observar um efeito tão extremo quanto o observado na amostra, caso a hipótese nula (de que a variável não tem efeito) seja verdadeira. Um valor-p menor que 0,05 (nível de significância adotado) indica que a variável é estatisticamente significativa, ou seja, é improvável que o efeito observado seja devido ao acaso. As variáveis com valor-p acima de 0,05 são consideradas não significativas e podem ser removidas do modelo para simplificá-lo e melhorar sua interpretabilidade.

Após essas transformações, o modelo de regressão múltipla final incluirá as seguintes variáveis independentes: *'artist_count'; 'released_month', 'released_day', 'in_spotify_playlists', 'in_spotify_charts', 'in_apple_playlists', 'in_deezer_playlists'* e *'in_shazam_charts'*. A variável dependente será o número de '*streams'*. Serão apresentados e discutidos na seção 3 (Resultados) as conclusões e diferenças entre a primeira e segunda versão do modelo mais detalhadamente.

2.3. Estimação do Modelo de Regressão

O modelo de regressão múltipla foi estimado utilizando o método de mínimos quadrados ordinários (OLS), que busca minimizar a soma dos quadrados dos resíduos, ou seja, a diferença entre os valores observados e os valores previstos pelo modelo. Para avaliar a qualidade do ajuste do modelo, foram utilizados o R-quadrado e o R-quadrado ajustado, que medem a proporção da variância da variável dependente (*streams*) explicada pelas variáveis independentes.

Além disso, o teste F foi aplicado para verificar a significância estatística global do modelo, ou seja, se pelo menos uma das variáveis independentes tem um efeito significativo sobre o número de *streams*. Os coeficientes de cada variável independente foram analisados, juntamente com seus respectivos valores-p, para determinar a significância estatística de cada uma delas no modelo.

A análise dos resíduos do modelo, incluindo a verificação da normalidade, homocedasticidade e ausência de autocorrelação, foi realizada para garantir que os pressupostos do modelo de regressão linear múltipla fossem atendidos. Gráficos de resíduos versus valores previstos e histogramas de resíduos foram utilizados para auxiliar nessa análise e estão presentes nos apêndices.

Os resultados da regressão, incluindo os coeficientes estimados, os valores de p, o R-quadrado, o R-quadrado ajustado, o teste F e os resultados dos testes de diagnóstico, serão apresentados e discutidos na seção 3 (Resultados). Essa discussão abordará a interpretação dos coeficientes, a significância estatística das variáveis e as implicações dos resultados para a compreensão dos fatores que influenciam a popularidade das músicas no Spotify.

3. RESULTADOS

3.1. Análise Diagnóstica

Após a estimação do modelo de regressão, realizamos uma análise diagnóstica para avaliar a adequação do modelo aos dados e verificar a presença de multicolinearidade. Para identificar multicolinearidade, utilizamos o Fator de Inflação da Variância (VIF), que quantifica o quanto a variância de um coeficiente estimado é aumentada devido à colinearidade entre as variáveis independentes, conforme apresentado no quadro abaixo:

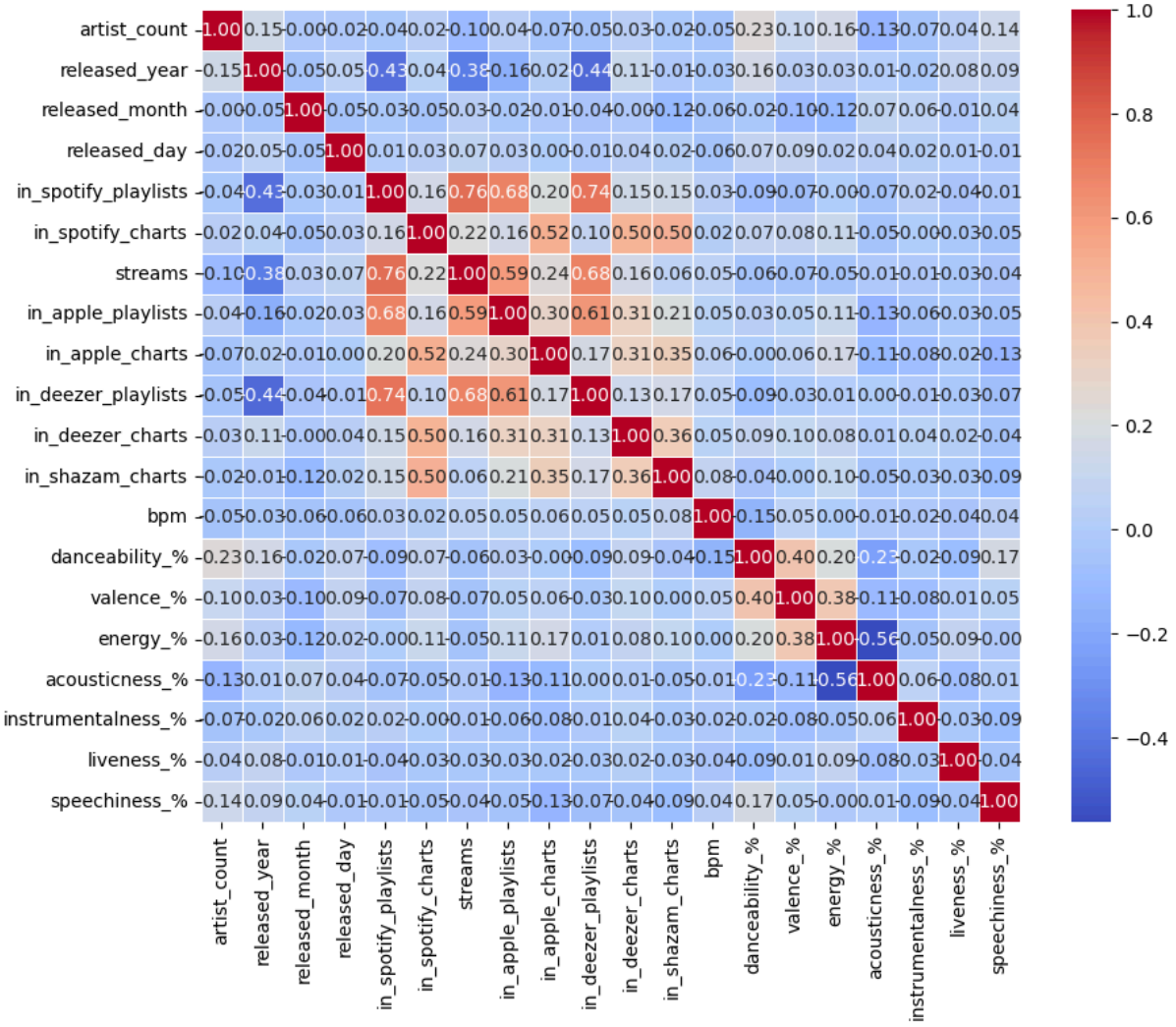
Nome da Variável	VIF
released_year	100.799.955
danceability_%	32.475.471
bpm	21.540.247
energy_%	29.409.201

Especificamente, identificamos que as variáveis *'released_year', 'bpm', 'danceability_%'* e *'energy_%'* apresentaram VIFs elevados e, portanto, foram excluídas do modelo. Esse processo aprimorou a estrutura dos dados e as relações entre as variáveis, permitindo uma especificação mais precisa do modelo.

A análise da matriz de correlação desempenhou um papel crucial na identificação de multicolinearidade, ou seja, a presença de fortes correlações entre as variáveis independentes. Essa etapa foi fundamental para evitar problemas na interpretação dos coeficientes e garantir a robustez do modelo de regressão. A partir dessa análise, foi possível identificar pares de variáveis com alta correlação e, estrategicamente, optar por incluir no modelo apenas uma delas, escolhendo aquela que apresentava maior relevância teórica e poder explicativo em relação à variável dependente '*streams*'. Esse processo de seleção criterioso de variáveis, baseado na matriz de correlação, contribuiu para a construção de um modelo mais parcimonioso,

com menor risco de instabilidade e maior capacidade de generalização para novas observações.

Matriz de Correlação das Variáveis Numéricas



Correlações Positivas Fortes:

- ***'released_year'* e *'in_spotify_playlists'*:** Músicas lançadas em anos mais recentes tendem a estar presentes em mais *playlists* do Spotify (correlação de 0.40). Isso sugere que a inclusão em *playlists* pode ser uma estratégia importante para aumentar a popularidade de músicas mais novas.
- ***'in_spotify_charts'* e *'in_spotify_playlists'*:** Músicas que aparecem em mais *charts* do Spotify também tendem a estar em mais *playlists* (correlação de 0.79). Essa forte correlação indica que essas duas variáveis estão intimamente relacionadas e podem ser usadas de forma intercambiável.

- ***'in_spotify_playlists'* e *'streams':*** Músicas presentes em mais *playlists* do Spotify tendem a ter um número maior de *streams* (correlação de 0.79). Essa é uma das correlações mais fortes encontradas na matriz, indicando que a presença em *playlists* é um fator crucial para o sucesso no Spotify.
- ***'in_spotify_charts'* e *'streams':*** Músicas que aparecem em mais *charts* do Spotify tendem a ter um número maior de *streams* (correlação de 0.25). Embora a correlação seja mais fraca do que a observada com in_spotify_playlists, ela ainda sugere que a presença em *charts* pode contribuir para a popularidade de uma música.
- ***'danceability_%'* e *'valence_%':*** Músicas com maior dançabilidade tendem a ser mais positivas (correlação de 0.41). Essa relação pode refletir a preferência dos ouvintes por músicas animadas e com energia positiva.

Correlações Negativas Fortes:

- ***'released_year'* e *'acousticness_%':*** Músicas lançadas em anos mais recentes tendem a ser menos acústicas (correlação de -0.18). Essa correlação sugere que as músicas mais novas tendem a ter um som mais eletrônico ou pop, em comparação com as músicas mais antigas.
- ***'energy_%'* e *'acousticness_%':*** Músicas com maior energia tendem a ser menos acústicas (correlação de -0.58). Essa é uma correlação negativa forte, indicando que a energia e a acústica são características musicais opostas.

Multicolinearidade:

- ***'released_year'* e *'released_month':*** A correlação perfeita (1.00) entre essas variáveis indica que uma é redundante em relação à outra. Isso significa que, para fins de análise, podemos excluir uma delas sem perder informações relevantes.
- ***'danceability_%', 'energy_%'* e *'valence_%':*** A forte correlação entre essas métricas de áudio (correlações entre 0.36 e 0.41) sugere que elas podem estar medindo aspectos semelhantes da música. Isso pode ser um problema para a análise de regressão, pois a multicolinearidade pode dificultar a interpretação dos coeficientes e a identificação das variáveis mais importantes.

A variável streams apresenta correlações moderadas com '*in_apple_playlists*' (0.77) e '*in_deezer_playlists*' (0.75), indicando que a presença em *playlists* nessas plataformas também pode influenciar o número de '*streams*'. No entanto, essas correlações são mais fracas do que a observada com '*in_spotify_playlists*', o que sugere que as *playlists* do Spotify podem ter um impacto maior na popularidade das músicas. Por outro lado as variáveis '*bpm*' (batidas por minuto) e '*key*' (tom)

apresentam correlações fracas com as demais variáveis, sugerindo que o ritmo e o tom da música podem não ter um impacto significativo no número de *streams*.

Considerando a análise exploratória, decidimos excluir a variável *'released_year'* devido à sua alta multicolinearidade com *'released_month'*. Além disso, optamos por excluir as variáveis *'in_apple_charts'* e *'in_deezer_charts'* por apresentarem baixa variância, o que significa que seus valores não variam muito e, portanto, não contribuem significativamente para explicar a variação no número de *streams*. Para lidar com a multicolinearidade, também decidimos excluir as variáveis *'danceability_%', 'energy_%'* e *'valence_%'.*

As variáveis com valor-p acima de 0,05 são consideradas não significativas foram removidas do modelo para simplificá-lo e melhorar sua interpretabilidade. Com base na análise da significância estatística, as variáveis não significativas excluídas do modelo foram: *'in_apple_charts', 'in_deezer_charts', 'bpm', 'danceability_%', 'valence_%', 'energy_%', 'acousticness_%', 'instrumentalness_%', 'liveness_%'* e *'speechiness_%'.*

Todo esse processo nos permitiu fazer a seleção das variáveis a serem incluídas no modelo, além de identificar e tratar os outliers e fazer a transformação de variáveis para garantir a adequação aos pressupostos do modelo de regressão linear múltipla. Após esse processo o modelo de regressão múltipla final incluiu as seguintes variáveis independentes: *'artist_count'; 'released_month', 'released_day', 'in_spotify_playlists', 'in_spotify_charts', 'in_apple_playlists', 'in_deezer_playlists'* e *'in_shazam_charts'.* A variável dependente será o número de *'streams'.*

3.2. Resultado

A análise de regressão múltipla buscou identificar os fatores que influenciam o número de reproduções de músicas ("*streams*"), inicialmente considerando todas as variáveis, exceto as que continham informações de identificação da música que não eram numéricas: *'artist(s)_name', 'artist_count', 'mode'* e *'key'.* O modelo estatístico construído revelou que 66,6% da variação nos streams pode ser explicada pelas variáveis incluídas, como o número de artistas na música, o dia de lançamento e a presença em playlists e charts de plataformas como Spotify, Apple Music e Deezer.

Entre as variáveis analisadas, algumas se destacaram como preditoras significativas do sucesso de uma música em termos de *streams*. A presença em playlists do Spotify, por exemplo, mostrou ter um impacto positivo substancial, enquanto um maior número de artistas na música foi associado a uma diminuição no número de streams. Características musicais como ritmo *'bpm'*, *'danceability_%'* e *'energy_%'* não apresentaram influência significativa nos resultados, conforme resultado abaixo:

OLS Regression Results

Dep. Variable:	streams	R-squared:	0.666
Model:	OLS	Adj. R-squared:	0.657
Method:	Least Squares	F-statistic:	72.47
Date:	Thu, 25 Jul 2024	Prob (F-statistic):	3.73e-150
Time:	12:48:00	Log-Likelihood:	-14567.
No. Observations:	711	AIC:	2.917e+04
Df Residuals:	691	BIC:	2.927e+04
Df Model:	19		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	3.144e+08	1.01e+08	3.100	0.002	1.15e+08	5.14e+08
artist_count	-9.273e+07	3.13e+07	-2.961	0.003	-1.54e+08	-3.12e+07
released_year	-2.345e+08	9.43e+07	-2.488	0.013	-4.2e+08	-4.94e+07
released_month	4.281e+07	2.35e+07	1.818	0.069	-3.41e+06	8.9e+07
released_day	8.166e+07	2.45e+07	3.337	0.001	3.36e+07	1.3e+08
in_spotify_playlists	1.234e+09	1.09e+08	11.326	0.000	1.02e+09	1.45e+09
in_spotify_charts	3.005e+08	4.99e+07	6.019	0.000	2.02e+08	3.99e+08
in_apple_playlists	2.191e+08	5.9e+07	3.714	0.000	1.03e+08	3.35e+08
in_apple_charts	5.541e+07	4.31e+07	1.286	0.199	-2.92e+07	1.4e+08
in_deezer_playlists	5.836e+08	8.11e+07	7.196	0.000	4.24e+08	7.43e+08
in_deezer_charts	-8.675e+06	4.62e+07	-0.188	0.851	-9.93e+07	8.19e+07
in_shazam_charts	-5.038e+08	7.71e+07	-6.532	0.000	-6.55e+08	-3.52e+08
bpm	5.014e+07	3.79e+07	1.324	0.186	-2.42e+07	1.24e+08
danceability_%	4.805e+07	4.4e+07	1.091	0.275	-3.84e+07	1.34e+08
valence_%	-4.94e+07	3.5e+07	-1.413	0.158	-1.18e+08	1.93e+07
energy_%	-8.559e+07	4.91e+07	-1.743	0.082	-1.82e+08	1.08e+07
acousticness_%	6.175e+06	3.54e+07	0.175	0.861	-6.33e+07	7.56e+07
instrumentalness_%	-8.343e+07	6.65e+07	-1.255	0.210	-2.14e+08	4.71e+07
liveness_%	2.068e+07	3.76e+07	0.550	0.582	-5.31e+07	9.45e+07
speechiness_%	-1.474e+07	3.08e+07	-0.479	0.632	-7.52e+07	4.57e+07

Omnibus:	200.292	Durbin-Watson:	1.915
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1138.535
Skew:	1.139	Prob(JB):	5.89e-248
Kurtosis:	8.766	Cond. No.	38.9

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Apesar do bom poder explicativo do modelo, alguns pontos merecem atenção. A distribuição dos resíduos não segue a normalidade esperada, o que pode comprometer a validade das inferências estatísticas. Além disso, a presença de multicolinearidade (alta correlação entre as variáveis preditoras) sugeriu a necessidade de refinamento do modelo, possivelmente através da remoção de variáveis redundantes. Para otimizar o modelo de regressão múltipla final incluímos apenas as seguintes variáveis independentes: *'artist_count'; 'released_month', 'released_day', 'in_spotify_playlists', 'in_spotify_charts', 'in_apple_playlists', 'in_deezer_playlists'* e *'in_shazam_charts';* além da variável dependente *'streams'*.

O segundo modelo, após a otimização e remoção das variáveis não significativas, apresenta algumas diferenças notáveis em relação ao primeiro, conforme resultado abaixo:

OLS Regression Results

Dep. Variable:	streams	R-squared:	0.657
Model:	OLS	Adj. R-squared:	0.653
Method:	Least Squares	F-statistic:	167.9
Date:	Tue, 23 Jul 2024	Prob (F-statistic):	1.95e-157
Time:	17:48:36	Log-Likelihood:	-14576.
No. Observations:	711	AIC:	2.917e+04
Df Residuals:	702	BIC:	2.921e+04
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	7.357e+07	2.01e+07	3.669	0.000	3.42e+07	1.13e+08
artist_count	-1.14e+08	2.99e+07	-3.809	0.000	-1.73e+08	-5.53e+07
released_month	5.292e+07	2.32e+07	2.276	0.023	7.28e+06	9.86e+07
released_day	7.346e+07	2.44e+07	3.016	0.003	2.56e+07	1.21e+08
in_spotify_playlists	1.315e+09	1.03e+08	12.812	0.000	1.11e+09	1.52e+09
in_spotify_charts	3.044e+08	4.18e+07	7.280	0.000	2.22e+08	3.87e+08
in_apple_playlists	1.862e+08	5.35e+07	3.480	0.001	8.11e+07	2.91e+08
in_deezer_playlists	6.386e+08	7.8e+07	8.183	0.000	4.85e+08	7.92e+08
in_shazam_charts	-4.919e+08	7.61e+07	-6.468	0.000	-6.41e+08	-3.43e+08

Omnibus:	207.123	Durbin-Watson:	1.921
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1140.044
Skew:	1.194	Prob(JB):	2.77e-248
Kurtosis:	8.725	Cond. No.	19.9

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Melhora na Interpretabilidade: O modelo otimizado é mais simples e direto, com apenas 8 variáveis preditoras em vez de 18. Isso facilita a interpretação dos resultados e a identificação dos principais fatores que influenciam os streams.

Manutenção do Poder Preditivo: Apesar da remoção de variáveis, o modelo otimizado mantém um alto poder preditivo, com um R-quadrado de 0.657, apenas ligeiramente menor que o R-quadrado de 0.663 do modelo original. Isso indica que as variáveis removidas não eram essenciais para explicar a variação nos streams.

Aumento da Significância Estatística: O modelo otimizado apresenta uma estatística F maior (167,9 vs. 75,59) e um p-valor menor (1.95e-157 vs. 8.84e-150), indicando que ele é ainda mais significativo estatisticamente do que o modelo original. Isso sugere que a remoção das variáveis não significativas melhorou a

capacidade do modelo de explicar a relação entre as variáveis preditoras e os streams.

Coeficientes Alterados: Os coeficientes das variáveis preditoras remanescentes no modelo otimizado são ligeiramente diferentes dos coeficientes do modelo original. Isso ocorre porque a remoção de variáveis pode alterar a forma como as variáveis restantes se relacionam com a variável dependente. No entanto, a direção e a magnitude dos efeitos permanecem consistentes.

O segundo modelo que foi otimizado apresentou-se uma melhor alternativa se comparado ao modelo inicial por ser mais simples, interpretável e estatisticamente mais significativo, sem sacrificar o poder preditivo. Ele fornece uma visão mais clara dos principais fatores que impulsionam os streams de música, o que pode ser útil para artistas, gravadoras e plataformas de streaming na tomada de decisões estratégicas.

4. CONCLUSÕES

Este estudo aprofundou a compreensão dos fatores que impulsionam a popularidade das músicas no Spotify, lançando luz sobre as variáveis que influenciam significativamente o número de streams. Através da análise de regressão múltipla, construímos um modelo estatístico capaz de explicar 66,6% da variação no número de streams. No entanto, a presença de multicolinearidade e a não normalidade dos resíduos indicaram a necessidade de otimização.

Após o refinamento, o modelo otimizado manteve um alto poder preditivo, explicando 65,7% da variação nos streams, com apenas 8 variáveis preditoras. Identificamos que a inclusão em playlists do Spotify (*'in_spotify_playlists'*) é um preditor crucial do sucesso, impactando positivamente o número de reproduções.

Curiosamente, o número de artistas envolvidos na produção musical (*'artist_count'*) demonstrou uma relação negativa com a popularidade, sugerindo que músicas com menos artistas tendem a ter mais streams. As variáveis *'released_month'* (mês de lançamento) e *'released_day'* (dia de lançamento) também se mostraram relevantes, indicando que o momento do lançamento pode desempenhar um papel estratégico no desempenho da música.

Em contraste, características musicais como ritmo ('bpm'), dançabilidade (*'danceability_%'*) e energia (*'energy_%'*) não apresentaram influência significativa no número de streams, desafiando a noção de que certos atributos musicais garantem automaticamente o sucesso.

Este estudo oferece insights valiosos para artistas, gravadoras e plataformas de streaming, fornecendo informações baseadas em dados para a tomada de decisões estratégicas. Ao entender as variáveis que realmente importam, os artistas podem otimizar suas estratégias de lançamento e promoção, enquanto as plataformas podem aprimorar seus algoritmos de recomendação e curadoria de playlists.

Em suma, este trabalho contribui para o crescente campo da análise de dados na indústria musical, demonstrando o potencial de modelos estatísticos para desvendar os segredos por trás do sucesso no streaming. Ao combinar dados e conhecimento musical, podemos capacitar os artistas a navegar com mais confiança no cenário digital em constante evolução, maximizando suas chances de alcançar um público amplo e construir carreiras musicais sólidas e duradouras.

AGRADECIMENTOS

Gostaria de expressar minha gratidão a todas as pessoas que contribuíram para a realização deste trabalho e para minha jornada acadêmica no Instituto Federal do Espírito Santo (IFES).

Primeiramente, gostaria de agradecer à minha esposa Ana Carolina, cujo apoio e encorajamento foram fundamentais para que eu ingressasse no IFES e realizasse um sonho de criança. Sua constante motivação foi um farol durante os momentos desafiadores dessa jornada.

Gostaria também de expressar minha sincera gratidão aos meus pais, sogros, irmãos e cunhados pelo constante apoio, encorajamento e compreensão ao longo desta jornada acadêmica.

Um agradecimento especial ao meu orientador, Prof. Guilherme Guilhermino, pela sua orientação, pela dedicação ao compartilhar seu conhecimento, disponibilidade e pelos valiosos insights ao longo deste trabalho. Sua expertise e orientação foram essenciais para o desenvolvimento deste estudo.

Agradeço também ao meu amigo Hugo F. Mauad, egresso do curso e parceiro de trabalho na empresa Blip, pela sua amizade, colaboração e troca de experiências ao longo da minha jornada profissional e acadêmica. Sua contribuição foi fundamental para o meu crescimento.

À comunidade de colegas do curso, expresso minha gratidão pelo bom humor, colaboração e motivação compartilhados ao longo desses anos. Por fim, gostaria de agradecer ao colegiado do curso por deliberar uma nova oferta de uma disciplina que fui impedido de cursar devido a uma licença médica decorrente de uma cirurgia

de emergência. Sem essa oportunidade, certamente não teria alcançado a conclusão deste curso. Sua compreensão e apoio foram essenciais para minha jornada acadêmica.

DECLARAÇÃO DE USO DE TECNOLOGIAS AUXILIADAS POR INTELIGÊNCIA ARTIFICIAL

Durante a elaboração deste trabalho, utilizei tecnologias de inteligência artificial (IA) que desempenharam papéis essenciais em diferentes etapas da pesquisa. Abaixo, listo as principais tecnologias de IA utilizadas e descrevo sua finalidade específica neste estudo:

OpenAI's GPT-3.5 Language Model: Utilizei o modelo de linguagem GPT-3.5 da OpenAI para auxiliar na geração de textos, revisão de conteúdo e fornecimento de insights durante a elaboração do trabalho. A IA contribuiu para a redação de seções específicas, como a revisão da literatura, a análise de resultados e a formulação de conclusões, fornecendo sugestões e aprimorando a qualidade do texto.

Gemini Pro: O modelo de linguagem Gemini Pro foi utilizado para refinar e aprimorar a redação do trabalho, incluindo a organização das ideias, a escolha de vocabulário e a estruturação das frases. A IA também auxiliou na identificação de possíveis erros gramaticais e ortográficos, contribuindo para a clareza e precisão do texto.

Em resumo, as tecnologias de inteligência artificial utilizadas neste trabalho desempenharam um papel fundamental na análise e interpretação dos dados, na redação e revisão do texto, contribuindo para uma pesquisa mais abrangente, precisa e informativa. A aplicação dessas tecnologias me permitiu explorar *insights* significativos, oferecendo uma visão aprofundada dos temas abordados e enriquecendo a qualidade e a relevância do trabalho apresentado.

REFERÊNCIAS

HAIR JR., Joseph F. et al. **Análise multivariada de dados**. 6. ed. Porto Alegre: Bookman, 2009.

MULLIGAN, Mark. **Music subscriber market shares 2023: New momentum**. MIDiA Research, [s.d.]. Disponível em: https://www.midiaresearch.com/blog/music-subscriber-market-shares-2023-new-momentum. Acesso em: 02 abr. 2024.

FRANCO, Pedro. **Mercado Fonográfico Brasileiro 2022**. Pro-Música Brasil, [s.d.]. Disponível em: https://pro-musicabr.org.br/home/numeros-do-mercado/. Acesso em: 02 abr. 2024.

ELGIRIYEWITHANA, Nidula. **Most Streamed Spotify Songs 2023.** Kaggle, [s.d.]. Disponível em: https://www.kaggle.com/datasets/nelgiriyewithana/top-spotify-songs-2023. Acesso em: 01 mar. 2024.

INSTITUTO FEDERAL
Espírito Santo
Campus Cariacica

APÊNDICES

Imagem 1 - Distribuição das Variáveis Numéricas

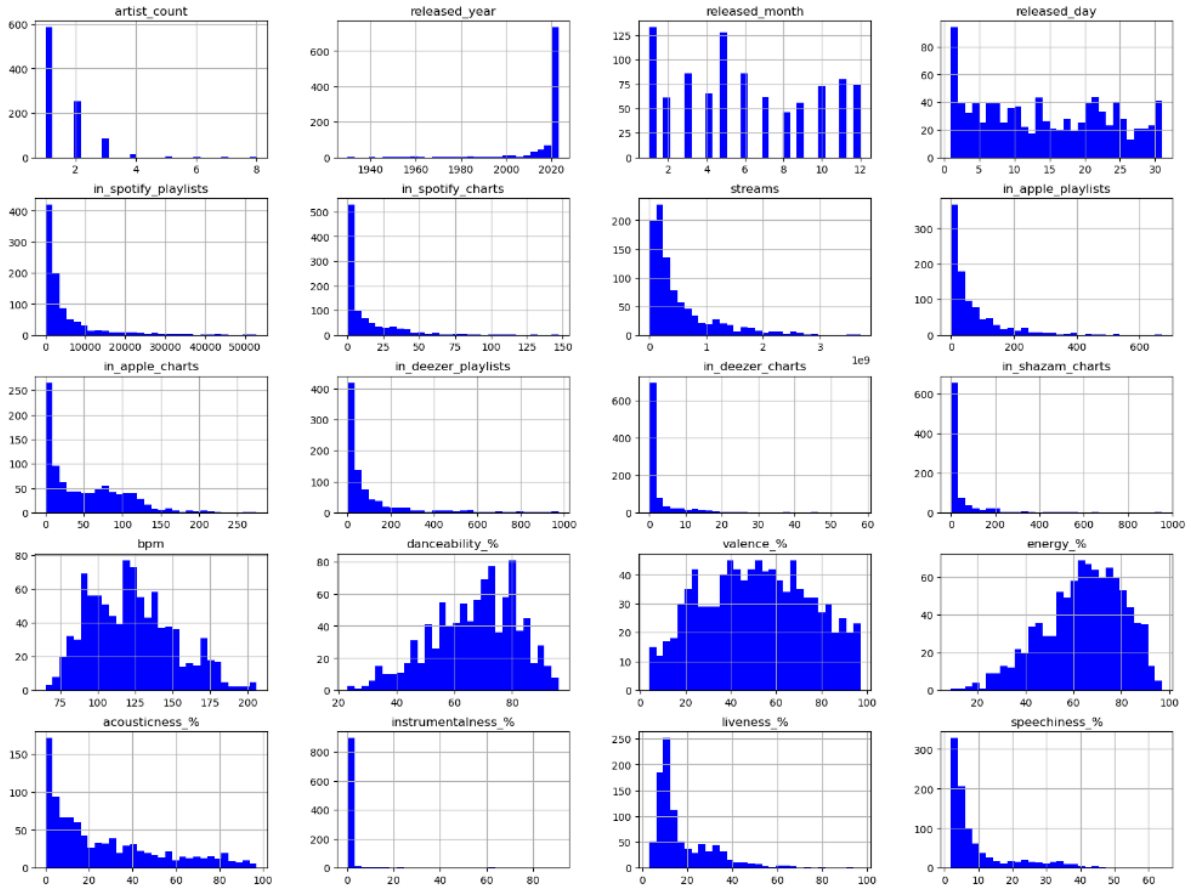


Imagem 2 - Gráficos Box Splot

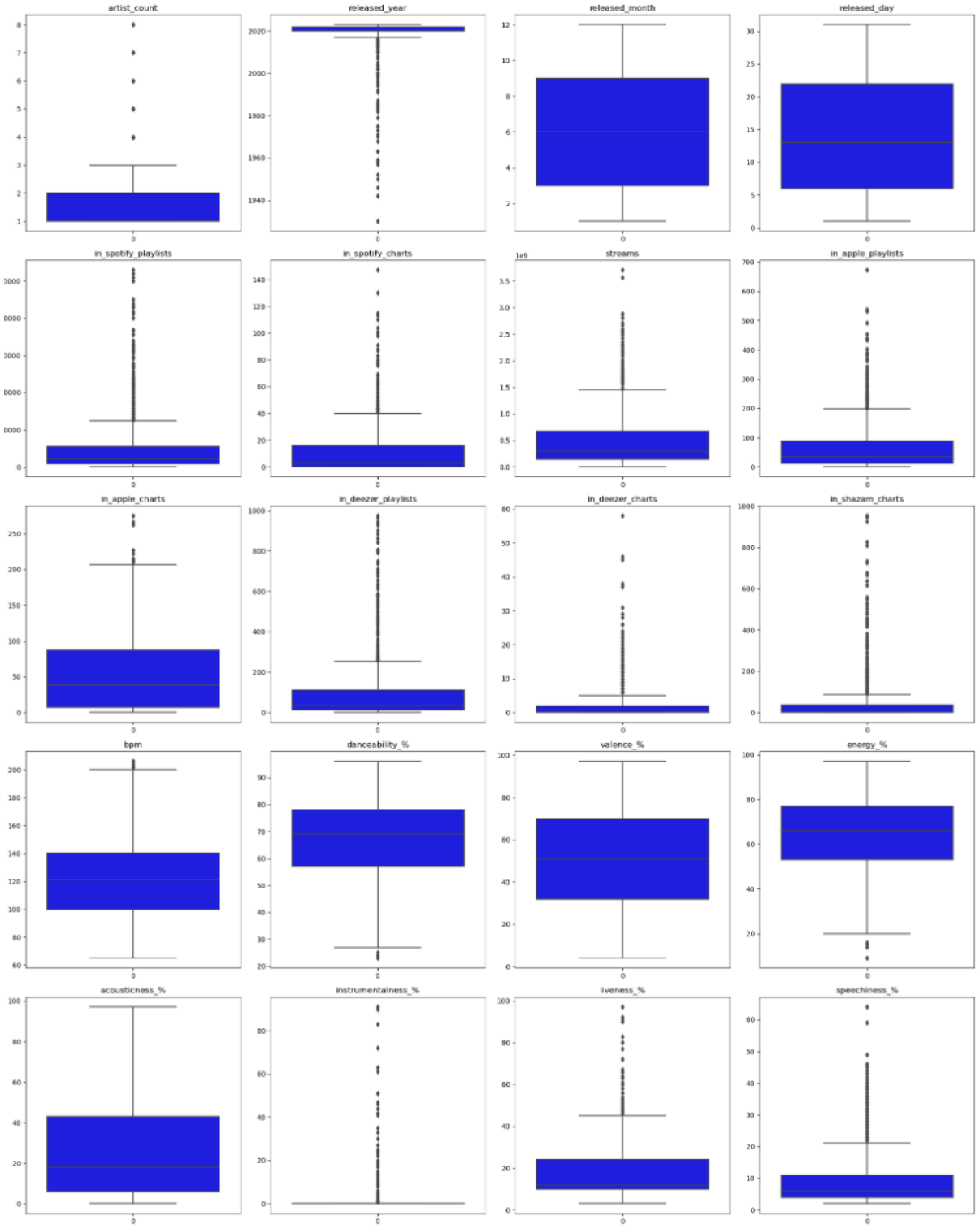


Imagem 3 - Distribuição de Resíduos

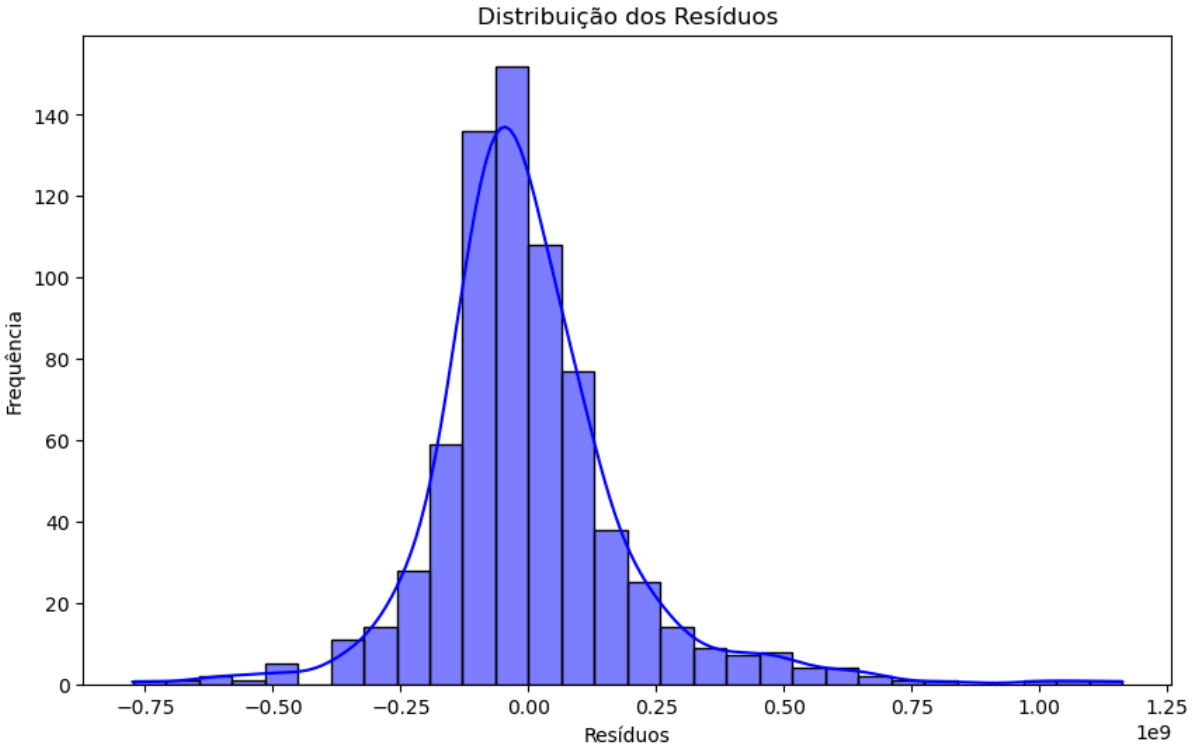
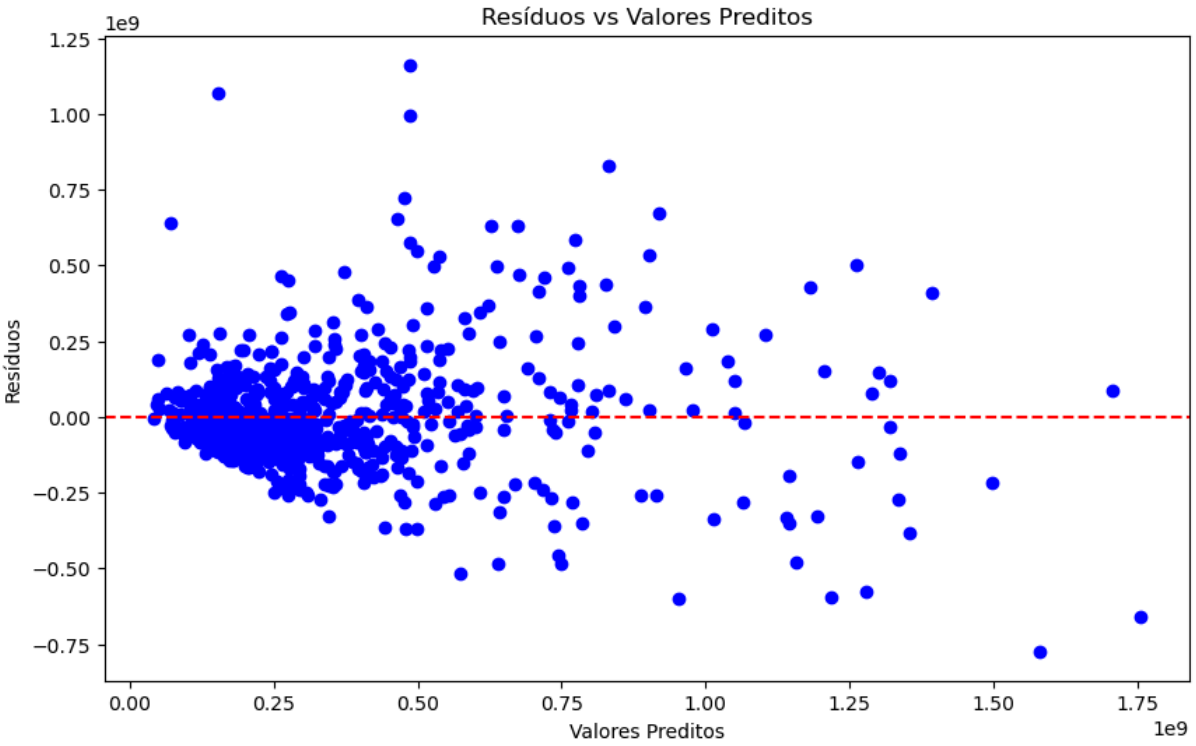


Imagem 4 - Valores Preditos



INSTITUTO FEDERAL
Espírito Santo
Campus Cariacica

Notebook - Script Phython

Análise Exploratória de Dados e Análise de Regressão Múltipla

Analista: Antonio Patricio

Data: 13/05/2024

Dataset: dataset1-charts_utf8.csv

Dataset_Url: https://www.kaggle.com/datasets/nelgiriyewithana/top-spotify-songs-2023

```
# Importar bibioteca Pandas para ler o arquivo CSV
import pandas as pd
```

```
# Carregar o conjunto de dados do arquivo CSV
data = pd.read_csv('abfss://59c7c841-9e0d-4096-8742-eee98785e02f@onelake.dfs.fabric.microsoft.com/4f3ddd72-657c-4ec2-acc1-2f283ab634bc/Files/dataset2-spotify-2023.csv')
# Visualizar as primeiras linhas do conjunto de dados do arquivo CSV
print("Visualização das primeiras linhas do conjunto de dados:")
print(data.head())
```

```
# Resumo estatístico das variáveis numéricas
print("\nResumo estatístico das variáveis numéricas:")
print(data.describe())
```

```
# Verificar o tipo de dados de cada coluna
print("\nTipos de dados de cada coluna:")
print(data.dtypes)
```

```
# Contar os valores únicos em cada coluna
print("\nContagem de valores únicos em cada coluna:")
print(data.nunique())
```

```
# Verificar valores ausentes em cada coluna
print("\nValores ausentes em cada coluna:")
print(data.isnull().sum())
```

```
# Remover linhas onde a coluna 'streams' não é numérica
data = data[pd.to_numeric(data['streams'], errors='coerce').notna()]
# Converter a coluna 'streams' para numérica removendo vírgulas
data['streams'] = data['streams'].str.replace(',', '').astype(float)
# Converter as colunas para o tipo de dados numérico apropriado
data['streams'] = pd.to_numeric(data['streams'], errors='coerce')
data['in_deezer_playlists'] = pd.to_numeric(data['in_deezer_playlists'], errors='coerce')
data['in_shazam_charts'] = pd.to_numeric(data['in_shazam_charts'], errors='coerce')
# Identificar as colunas de tipo int64 e float64
colunas_int_float = data.select_dtypes(include=['int64', 'float64']).columns
# Selecionar apenas as colunas de tipo int64 e float64
data_int_float = data[colunas_int_float]
# Verificar o DataFrame resultante
```

```
print(data_int_float)
# Converter apenas as colunas específicas para float
data = data_int_float.astype(float)
# Verificar novamente os tipos de dados
print(data.dtypes)
```

```
# Importar bibiotecas adicionais
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
# Gerar histogramas
data.hist(bins=30, figsize=(20, 15), color='blue')
plt.suptitle('Distribuição das Variáveis Numéricas', fontsize=20)
plt.show()
# Gerar box plots
# Número de colunas no DataFrame
num_cols = len(data.columns)
# Determinar o número de linhas e colunas para o grid de subplots
num_rows = (num_cols // 4) + (num_cols % 4 > 0)
num_cols_plot = min(num_cols, 4)
plt.figure(figsize=(20, num_rows * 5))  # Ajuste o tamanho da figura conforme necessário
for i, column in enumerate(data.columns, 1):
    plt.subplot(num_rows, num_cols_plot, i)
    sns.boxplot(data=data[column], color='blue')
    plt.title(column)
plt.tight_layout()
plt.show()
```

```
# Função para detectar outliers usando o método do IQR
def detectar_outliers_iqr(data):
    outliers = pd.DataFrame()
    for column in data.select_dtypes(include=[np.number]).columns:
        Q1 = data[column].quantile(0.25)
        Q3 = data[column].quantile(0.75)
        IQR = Q3 - Q1
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR
        outliers[column] = ((data[column] < lower_bound) | (data[column] > upper_bound))
    return outliers
# Identificar outliers
outliers_iqr = detectar_outliers_iqr(data)
# Mostrar outliers
print("Outliers detectados pelo método do IQR:")
print(outliers_iqr)
# Remover outliers do DataFrame
data_sem_outliers = data[~outliers_iqr.any(axis=1)]
```

```
# Importar bibiotecas adicionais
from scipy import stats
# Função para detectar outliers usando o Z-Score
def detectar_outliers_zscore(data, threshold=3):
    outliers = pd.DataFrame()
    for column in data.select_dtypes(include=[np.number]).columns:
        z_scores = np.abs(stats.zscore(data[column]))
        outliers[column] = z_scores > threshold
    return outliers
# Identificar outliers
outliers_zscore = detectar_outliers_zscore(data)
```

```
# Mostrar outliers
print("Outliers detectados pelo método do Z-Score:")
print(outliers_zscore)
# Remover outliers do DataFrame
dtata_sem_outliers_zscore = data[~outliers_zscore.any(axis=1)]
```

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
# Suponha que df seja o seu DataFrame com as variáveis independentes
# Gerar a matriz de correlação
correlation_matrix = dtata_sem_outliers_zscore.corr()
# Exibir a matriz de correlação
print(correlation_matrix)
# Analisar a matriz de correlação para identificar pares de variáveis com alta correlação
# (Correlações acima de 0.8 ou abaixo de -0.8)
threshold = 0.8
highly_correlated_pairs = []
for i in range(len(correlation_matrix.columns)):
    for j in range(i):
        if abs(correlation_matrix.iloc[i, j]) > threshold:
            pair = (correlation_matrix.columns[i], correlation_matrix.columns[j], correlation_matrix.iloc[i, j])
            highly_correlated_pairs.append(pair)

print("Pares de variáveis com alta correlação (acima de 0.8 ou abaixo de -0.8):")
for pair in highly_correlated_pairs:
    print(pair)
# Tratar valores ausentes e infinitos
df = dtata_sem_outliers_zscore.replace([np.inf, -np.inf], np.nan)
df = dtata_sem_outliers_zscore.dropna()
# Calcular o VIF para cada variável
X = df.values
vif = pd.DataFrame()
vif["Variável"] = df.columns
vif["VIF"] = [variance_inflation_factor(X, i) for i in range(X.shape[1])]
print("Fatores de Inflação da Variância (VIF):")
print(vif)
# Identificar variáveis com VIF superior a 10
high_vif = vif[vif["VIF"] > 10]
print("Variáveis com VIF superior a 10, indicando multicolinearidade alta:")
print(high_vif)
```

```
# Visualizar a matriz de correlação das variáveis numéricas
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
plt.title('Matriz de Correlação das Variáveis Numéricas')
plt.show()
```

```
# Verificar se há valores ausentes e substituir
if df.isnull().values.any():
    print("O conjunto de dados contém valores ausentes. Imputando valores...")
    df.fillna(0, inplace=True)  # Preencher valores ausentes com 0 (ou outro valor apropriado)
# Verificar se há valores infinitos em colunas numéricas
numeric_columns = df.select_dtypes(include=[np.number]).columns
inf_values = df[numeric_columns].apply(lambda x: np.isinf(x)).any()
if inf_values.any():
```

```
    print("O conjunto de dados contém valores infinitos em colunas numéricas. Substituindo...")
    df.replace([np.inf, -np.inf], np.nan, inplace=True)
    df.dropna(inplace=True)  # Remover linhas com valores infinitos
print("Pronto!")
```

```
# Importar bibiotecas adicionais
import statsmodels.api as sm
dadofinal = pd.DataFrame(df)
print(dadofinal)
```

```
import statsmodels.api as sm
from sklearn.preprocessing import MinMaxScaler
# ... (seu código para criar o dadofinal DataFrame) ...
# Definir as variáveis independentes (X) e a variável dependente (y)
X = dadofinal[["artist_count","released_year","released_month","released_day","in_spotify_playlists","in_spotify_charts","in_apple_playlists","in_apple_charts","in_deezer_playlists","in_deezer_charts","in_shazam_charts","bpm", "danceability_%","valence_%","energy_%","acousticness_%","instrumentalness_%","liveness_%","speechiness_%"]]
y = dadofinal["streams"]
# Normalizar os dados
scaler = MinMaxScaler()
X_normalized = scaler.fit_transform(X)
# Adicionar uma constante à matriz X para ajustar o termo de interceptação
X_normalized = sm.add_constant(X_normalized)
# Criar um DataFrame com os dados normalizados para manter os nomes das colunas
X_normalized_df = pd.DataFrame(X_normalized, columns=['const'] + X.columns.tolist())
# Ajustar os índices de y e X_normalized_df para garantir que estejam alinhados
y = y.reset_index(drop=True)
X_normalized_df = X_normalized_df.reset_index(drop=True)
# Ajustar o modelo de regressão
modelo = sm.OLS(y, X_normalized_df).fit()
# Imprimir os resultados
print(modelo.summary())
```

```
# Verificar as colunas P > |t|
p_values = modelo.pvalues[1:]  # Ignoramos o valor constante
insignificantes = p_values[p_values > 0.05]  # Selecionamos as variáveis com p-value > 0.05
print("Variáveis não significativas:")
print(insignificantes)
```

```
# Definir as variáveis independentes (X) e a variável dependente (y)
X = dadofinal[["artist_count","released_month","released_day","in_spotify_playlists","in_spotify_charts",
"in_apple_playlists","in_deezer_playlists","in_shazam_charts"]]
y = dadofinal["streams"]
# Normalizar os dados
scaler = MinMaxScaler()
X_normalized = scaler.fit_transform(X)
# Adicionar uma constante à matriz X para ajustar o termo de interceptação
X_normalized = sm.add_constant(X_normalized)
# Criar um DataFrame com os dados normalizados para manter os nomes das colunas
```

```
X_normalized_df = pd.DataFrame(X_normalized, columns=['const'] + X.columns.tolist())
# Ajustar os índices de y e X_normalized_df para garantir que estejam alinhados
y = y.reset_index(drop=True)
X_normalized_df = X_normalized_df.reset_index(drop=True)
# Ajustar o modelo de regressão
modelo_simplificado = sm.OLS(y, X_normalized_df).fit()
# Imprimir os resultados
print(modelo_simplificado.summary())
```

```
# Obtendo os resíduos do segundo modelo
residuos = modelo_simplificado.resid
# Visualizando a distribuição dos resíduos
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
sns.histplot(residuos, kde=True, color='blue', bins=30)
plt.title('Distribuição dos Resíduos')
plt.xlabel('Resíduos')
plt.ylabel('Frequência')
plt.show()
# Verificando a homocedasticidade dos resíduos
plt.figure(figsize=(10, 6))
plt.scatter(modelo_simplificado.fittedvalues, residuos, color='blue')
plt.title('Resíduos vs Valores Preditos')
plt.xlabel('Valores Preditos')
plt.ylabel('Resíduos')
plt.axhline(y=0, color='red', linestyle='--')
plt.show()
```

Antonio Patricio Queres Neto

HARMONIZANDO COM O PÚBLICO: UM ESTUDO DAS VARIÁVEIS QUE IMPACTAM A POPULARIDADE DE MÚSICAS DO SPOTIFY

Trabalho Final de Curso apresentado à Coordenadoria do Curso de Engenharia de Produção do Instituto Federal do Espírito Santo – *campus* Cariacica como requisito parcial para obtenção do título de Especialista em Engenharia de Produção com Ênfase em Ciência de Dados

Aprovado em 05 de julho de 2024

COMISSÃO EXAMINADORA

Prof. Guilherme Guilhermino Neto, D.Sc.

Ifes – Instituto Federal do Espírito Santo

Orientador

Prof. Erivelto Fioresi de Sousa, D.Sc.

Ifes – Instituto Federal do Espírito Santo

Membro da banca avaliadora

Prof. Pedro Matos da Silva, D.Sc.

Ifes – Instituto Federal do Espírito Santo

Membro da banca avaliadora

FOLHA DE APROVAÇÃO-TCC Nº 18/2024 - CAR-CCEP (11.02.19.01.08.03.10)

(Nº do Protocolo: NÃO PROTOCOLADO)

(Assinado digitalmente em 30/07/2024 17:01)
ERIVELTO FIORESI DE SOUSA
PROFESSOR DO ENSINO BASICO TECNICO E TECNOLOGICO
CAR-CCE (11.02.19.01.08.03.11)
Matrícula: 1579284

(Assinado digitalmente em 30/07/2024 16:54)
GUILHERME GUILHERMINO NETO
PROFESSOR DO ENSINO BASICO TECNICO E TECNOLOGICO
CAR-CCEP (11.02.19.01.08.03.10)
Matrícula: 2151589

(Assinado digitalmente em 31/07/2024 20:46)
PEDRO MATOS DA SILVA
PROFESSOR DO ENSINO BASICO TECNICO E TECNOLOGICO
CAR-CCTA (11.02.19.01.08.03.02)
Matrícula: 2460822

*Visualize o documento original em https://sipac.ifes.edu.br/documentos/ informando seu número: **18**, ano: **2024**, tipo: **FOLHA DE APROVAÇÃO-TCC**, data de emissão: **30/07/2024** e o código de verificação: **72169a15a4***