UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE CIÊNCIAS - CAMPUS BAURU
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LUCAS PONTE CORREIA

ANÁLISE DE DADOS PARA ESTUDO DA RELAÇÃO ENTRE TENDÊNCIAS MUSICAIS E FENÔMENOS SOCIO-ECONÔMICOS

Nome: Lucas Ponte Correia	RA: 141023601

Orientador: Clayton Reginaldo Pereira	Assinatura:

BAURU
2018

LUCAS PONTE CORREIA

ANÁLISE DE DADOS PARA ESTUDO DA RELAÇÃO ENTRE TENDÊNCIAS MUSICAIS E FENÔMENOS SOCIO-ECONÔMICOS

Proposta para Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.
Orientador: Prof. Dr. Clayton Reginaldo Pereira

BAURU
2018

Lucas Ponte Correia

Análise de dados para estudo da relação entre tendências musicais e fenômenos socio-econômicos

Proposta para Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.

Banca Examinadora

Prof. Dr. Clayton Reginaldo Pereira
Orientador

Profª Dra. Simone das GraçasDomingues Prado

Bauru, _____ de _____ de _____.

Agradecimentos

Agradeço a todos os amigos que tiveram mais fé em mim e nesse projeto do que eu mesmo, e fortaleceram minha determinação de concluí-lo. O meu orientador, Prof^{o} Clayton Reginaldo Pereira, pela orientação e por aceitar ser parte desse projeto. A minha família, que me apoiou em todas as etapas desse trabalho e curso. Por fim, a meu Deus, por me sustentar até esse momento.

Resumo

Música e emoções humanas são intimimamente ligadas, uma podendo afetar a outra. Do mesmo modo, as circustâncias em que uma pessoa se encontra economica e socialmente também é um fator no estresse e saúde mental das pessoas. O presente projeto pretende explorar se, baseado nessas correlações, podemos encontrar padrões nas músicas mais ouvidas, especialmente em plataformas digitais, baseando-se na situação econômica de uma região.
Palavras-chave: Música, emoções, economia, indicadores.

Abstract

Music and human emotions are closely connected, whereas one may affect the other. In a similar manner, an individual's current economic or social circumstances are deciding factor in one's stress and mental health. This project intends to explore if, based on the aforementioned correlations, we may find pattens in the current most popular songs, specially in digital plataforms, based on a region's economic situation. **Keywords:** Music, emotions, economy, indicators.

Lista de ilustrações

Figura 1 – Visualização do arquivo CSV gerado. 14
Figura 2 – Visualização do arquivo CSV após tratamento. 15
Figura 3 – Resultados de predição de todos os atributos de áudio. 26
Figura 4 – Resultados de predição da valência. 27
Figura 5 – Resultados de predição da valência baseado nos demais atributos das faixas. 28

inserir lista de tabelas

Lista de tabelas

Tabela 1 – Descrição das características da análise de áudio 22
Tabela 1 – Descrição das características da análise de áudio 23
Tabela 1 – Descrição das características da análise de áudio 24
Tabela 2 – Métricas de avaliação do algoritmo de regressão. 24
Tabela 2 – Métricas de avaliação do algoritmo de regressão. 25
Tabela 3 – Resultados de predição de todos os atributos de áudio. 26
Tabela 4 – Resultados de predição da valência. 27
Tabela 5 – Resultados de predição da valência. 28

Lista de códigos

1 Script de mesclagem entre os dados da análise de áudio e indicadores econômicos 18

Sumário

Lista de códigos **8**

1 INTRODUÇÃO **10**
1.1 Problema **11**
1.2 Justificativa **11**
1.3 Objetivos **11**
1.3.1 Objetivo Geral 11
1.3.2 Objetivos Específicos 11

2 PROJETO E IMPLEMENTAÇÃO **13**
2.1 Ferramentas utilizadas **13**
2.2 Coleta, seleção e tratamento inicial dos dados **13**
2.2.1 Extração das músicas mais populares 13
2.2.2 Coleta de indicadores econômicos 15
2.2.3 Análise de recursos de áudio 16
2.2.4 União dos dados de áudio com os *sets* econômicos 17
2.3 Algoritmo de análise **20**
2.3.1 Pré processador 20
2.3.2 Visualização dos resultados 22
2.4 Experimentação e resultados **22**

3 CONCLUSÃO **29**

REFERÊNCIAS **30**

1 Introdução

Música é uma parte integral da cultura humana de uma maneira quase universal, e diversos estudos até mesmo apontam uma forte correlação com a psique e as emoções (KRUMHANSL, 2002), tanto como influenciadora de comportamento e sentimentos como influenciada pelos mesmos em sua construção e consumo. Desse modo, é fácil deduzir que fatores externos que também afetam essas áreas do nosso psicológico indiretamente afetam o nosso comportamento como ouvintes e produtores musicais. Desse modo, podemos extender essa conexão comportamental e propor que tendências de composição e popularidade de peças podem ser usadas para medir e entender esses fenômenos econômicos e sociais.

Com o avanço da tecnologia da informação e comunicação surgiu não só a nescessidade de processar dados de maneira global e encontrar métodos mais inteligentes e sofisticados de não apenas processá-los mas também compreender seus paralelos como também veio a demanda e a popularização de serviços *online* para suprir por meio de um mercado virtual ações cotidianas, especialmente na esfera do entretenimento, entre elas assistir vídeos, conversar ou manter contato com pessoas remotas e, naturalmente, ouvir música (ALJANAKI; YANG; SOLEYMANI, 2015).

A esse grande conjunto de dados imprecisos, e por extensão ferramentas, técnicas de processamento e manipulação, e aplicações associadas em seu estudo, denominamos *big data* (SINGH et al., 2015). Até avanços maiores nas áreas de comunicação e inteligência pouco se desenvolvia sobre o assunto, dado que a estrutura comum de armazenamento de dados (como bancos de dados) era suficiente para a maior parte das aplicações e que, sem o uso de técnicas de aprendizagem e *hardware* mais robusto, o processamento de tal quantidade de dados seria inviável. Entretanto, a maior integração entre computadores, serviços e aplicações, com a maior democratização da *internet* e o adventos dos dispositivos *mobile*, bem como evoluções nos campos teórico e prático de IA, viu-se a oportunidade de extender esse campo, o que resultou em não somente soluções práticas para problemas do mundo real, como também em métodos de processamento e mineração de dados também aplicáveis a nível funcional para conjuntos menores de dados.

Como os serviços mencionados acima e eventos de ordem econômico-social geram um grande volume de dados (ALJANAKI; YANG; SOLEYMANI, 2015) que pode ser acessado para análise, pode-se usar técnicas de análise de dados provenientes do *big data* para encontrar as relações propostas no primeiro parágrafo entre comportamento de consumo e produção musical e eventos que afetam a sociedade em maior escala. Esse trabalho assim se propõe a utilizar algoritmos e ferramentas de *big data* e *data mining* para montar um *software* que processe e parseie esses dados a fim de determinar se pode-se concluir uma relação definida entre eventos

globais e locais e tendências no mundo da música.

1.1 Problema

Conforme discorrido, é conhecido que música gera uma resposta emocional no ouvinte e assim reflete muito sobre seu contexto, de modo que é comumente usada como objeto de estudo para melhor entender a época e o contexto socioeconômico de sua composição. O que se deseja aferir é uma relação mais indireta entre comportamentos relacionados à música, sua produção e seu consumo e o contexto social, político e econômico de um determinado local, de modo a saber se é válido, a partir da análise destes, fazer predições sobre aquele e vice-versa.

Com a popularização dos serviços de *streaming* de música (como *Spotify*, *Deezer* e outros) e maior acesso a *data sets* que tratam da situação econômica de uma região em um certo tempo, torna-se viável o uso de técnicas de mineração e análise de dados como ferramentas para buscar correlações entre o ambiente econômico e os tipos de peças que são compostas ou mais populares.

1.2 Justificativa

Como citado, os processos de análise do contexto econômico e social ainda possuem baixa implementação de processos de automatização, o que consome tempo de pesquisa que poderia ser otimizado com ferramentas de análise de dados que auxiliassem na extração de dados e correlações significativas para o campo de estudo.

A motivação do presente trabalho é a exploração da correlação entre a música em seu contexto popular e prover uma ferramenta que possa auxiliar na extração de significado para campos de pesquisa mais afastados da área de tecnologia, que muitas vezes não se valem de recursos desse tipo.

1.3 Objetivos

1.3.1 Objetivo Geral

Desenvolver um *software* de extração e análise de dados que permita traçar correlações entre um cenário econômico e a tipografia de suas músicas populares.

1.3.2 Objetivos Específicos

- Estudar sobre análise, mineração e redução de dados para datasets maiores;
- Desenvolver o *software* de redução e análise;

- Alimentar o produto com dados provenientes de fontes variadas;
- Averiguar a precisão das correlações traçadas para saber se são relevantes o bastante para ser uma ferramenta de apoio;

2 Projeto e Implementação

2.1 Ferramentas utilizadas

A linguagem de programação escolhida para o desenvolvimento do trabalho foi o *Python*. A escolha se deu pelo seu ecossitema rico no setor de *data science*, contando com diversos *plugins*, pacotes e bibliotecas para manipulação de dados e *machine learning*; e por sua facilidade de uso e aprendizado. Dentre as ferramentas usadas juntamente com a liguagem base estão: o *Pandas*, biblioteca de estruturas e manipulação de dados, facilitando o uso e tratamento dos *datasets*; o *scikit-learn*, um módulo *Python* que integra vários algoritmos de aprendizado de máquina estado da arte para problemas supervisionados ou não de até média escala (PEDREGOSA et al., 2011); A biblioteca *scikit-optimize*, que expande os métodos do *scikit-learn* com fins de otimização; a biblioteca *spotipy* (LAMERE, 2019), para fornecer uma interface de comunicação mais simples com a API (*application programming interface*) oficial do *Spotify*; e um *web crawler*, *software* que atravessa e realiza o *download* de documentos *web* de modo automático e metódico (KAUSAR; DHAKA; SINGH, 2013), chamado *Spotify Charts API* (GITHUB, 2019a), que conta com funções para, através de chamadas *http* obter as informações das músicas listadas da página de *charts* oficial da plataforma (SPOTIFY AB, 2019b), podendo agrupá-las ou organizá-las de acordo com intervalo de tempo e país.

2.2 Coleta, seleção e tratamento inicial dos dados

2.2.1 Extração das músicas mais populares

O primeiro passo do projeto consistiu em selecionar os conjuntos de dados a serem trabalhados. Conforme mencionado anteriormente, os dados foram selecionados a partir de serviços de *streaming* por sua crescente popularidade e maior acessibilidade aos dados. Para os fins deste projeto, foi selecionada a plataforma *Spotify*. O foco principal da lista, dada a natureza do projeto, deveria ser uma representação fiel, ou ao menos aproximada, do conjunto de peças mais populares em um determinado período de tempo, com o qual poderiam ser cruzados dados de índices econômicos.

Inicialmente, foram pesquisadas *sets* de dados prontos, listando músicas por ordem de popularidade. Entretanto, nenhuma base foi encontrada com uma quantidade significativa de dados relevantes ou, mais importante, com informações sobre os períodos de tempo e regiões em que foram mais populares, informações cruciais para comparar com indicadores econômicos, que também variam de acordo com essas grandezas. A solução encontrada foi extrair os dados por meio de *crawlers* que acessassem as bases de dados do *Spotify* baseando-

se na sua popularidade. Inicialmente considerou-se utilizar os recursos de pesquisa da API oficial da plataforma (SPOTIFY AB, 2019a) para construir um *script* para a obtenção dos dados, mas após maior estudo optou-se por extrair os dados a partir de um outro recurso oficial: os *charts* ou as "paradas"oficiais (SPOTIFY AB, 2019b). Essa página consiste em uma lista, atualizada diariamente, das 200 (opcionalmente, as 50) músicas mais tocadas da plataforma em uma determinada data, e conta com diversos recursos interessantes ao projeto, os principais destes sendo o filtro por país/região (selecionando as músicas mais reproduzidas no lugar selecionado) e a capacidade de selecionar os dias para referência.

Para acessar os dados, valeu-se de uma versão modificada (de minha autoria) do *Spotify Charts API* (GITHUB, 2019b). As modificações no código do mesmo foram para otimização do tempo de coleta, melhor adequação ao ambiente e correção de bugs na execução. Com isso feito, foi construído um importador integrando o pacote descrito, e uma interface em texto simples para determinar os parâmetros básicos e disparar a extração de dados, gerando arquivos em formato CSV com as listas, agrupadas por ano (isto é, um arquivo para músicas no charts de 2017, outro para as de 2018, etc.). Nessa extração, foi considerada apenas a região do Brasil. Devido a limitações na plataforma, só eram acessíveis dados a partir da segunda metade de 2016, mas o volume de dados gerados para os anos 2017 e 2018 (em torno de 73000 entradas por arquivo, correspondendo às 200 músicas multiplicadas pelos dias do ano) foi considerado suficiente para os fins deste projeto.

Os dados extraídos da página foram: *ranking* da faixa na lista, o seu nome, o nome do artista ou artistas, o seu número de reproduções na plataforma até a presente data, a data e região de referência da lista, e por fim o código identificador da música no serviço do *Spotify*. A figura 1 mostra com maior detalhe a tabela gerada.

Figura 1 – Visualização do arquivo CSV gerado.

	A	B	C	D	E	F	G
1	Position	Track Name	Artist	Streams	date	region	id
2	1	Vai malandra (feat. Tropkillaz & DJ Yuri Martins)	Anitta	1060057	2018-01-01	br	6u0EAxf1OJTLS7CvlnuNd7
3	2	Agora Vai Sentar	MC's Jhowzinho & Kadinho	524222	2018-01-01	br	0pDaqqlForVNO4jrtTxcWT
4	3	Amar Amei	Mc Don Juan	427165	2018-01-01	br	1kNVJQEkobOlyfbctPZ4fs
5	4	Ar-Condicionado No 15 - Ao Vivo	Wesley Safadão	426318	2018-01-01	br	5ac0YTpOaDX9wEYYyGCgIL
6	5	Downtown	Anitta	409942	2018-01-01	br	3Ga6eKrUFf12ouh9Yw3v2D
7	6	Rabiola	MC Kevinho	404255	2018-01-01	br	228vRHzGfQh47EfLjuY1Yr
8	7	Deixa Ela Beijar	Matheus & Kauan	395307	2018-01-01	br	7q5agbUOzZnVF4Yes3AD7I
9	8	Regime Fechado - Ao Vivo	Simone & Simaria	362771	2018-01-01	br	6UZJOc2HBrTp9DYfir9N8n
10	9	Fazer Falta	Mc Livinho	358240	2018-01-01	br	6pSYjx66rlqRmGGTHhnjCo
11	10	Permanecer	Lucas Lucco	347521	2018-01-01	br	3sIMCkwUZbQbWeNOTG3E
12	11	Ritmo Mexicano	Mc Gw	344122	2018-01-01	br	6ZR7rT7vlwtY0EqyLeyAaO
13	12	Encaixa	MC Kevinho	332782	2018-01-01	br	7yYOMPwpV5CsK0cxoAZT6
14	13	Raspão - Ao Vivo	Henrique & Diego	316418	2018-01-01	br	6gquljDBW7rE6mCBzmTRxo
15	14	Vidinha de Balada - Ao Vivo	Henrique & Juliano	315514	2018-01-01	br	2g2ZjM1GjY9Lf2Ff0L94Ul
16	15	Corpo Sensual (feat. Mateus Carrilho)	Pabllo Vittar	296590	2018-01-01	br	4kkQGhScJDzNIlGawQE9SN
17	16	Big Jet Plane	Alok	294098	2018-01-01	br	1xd3OqTbx17HutN7RfJOTj
18	17	Contrato	Jorge & Mateus	293940	2018-01-01	br	12IOxvml5EBvSCzQll7xVS
19	18	Abusadamente	MC Gustta	286731	2018-01-01	br	7vgNGxchhqDQWjKjLPTPro
20	19	Sua cara (feat. Anitta & Pabllo Vittar)	Major Lazer	285290	2018-01-01	br	3ibRxEv4KkcU9e259eyNJA
21	20	Cê Acredita - Ao Vivo	João Neto & Frederico	272033	2018-01-01	br	5JZIDZzWPbrYwMHmnOQO
22	21	A Mala É Falsa - Ao Vivo	Felipe Araújo	268868	2018-01-01	br	6BPt2sVRvE41rUixcCAHY6
23	22	Apelido Carinhoso	Gusttavo Lima	258436	2018-01-01	br	6wXYGqXCLtEHx4VDb2TPcM

Fonte: Elaborada pelo autor.

As listas geradas contém um número considerável de tuplas, o que pode ser um problema para algoritmos de aprendizado mais modestos. Mais do que isso, a maior parte destas

consistiam em entradas redundantes, já que as faixas levam um tempo significante para sair de seu pico de popularidade e, portanto, do *index* de músicas mais populares usado para extrair os dados. Assim, foi escrito outro *script* com o objetivo de limpar as entradas de dados repetidas e já cortar informações irrelevantes ao contexto do projeto, como por exemplo nome do artista. O *script* foi executado sobre as tabelas dos dois anos pesquisados (2017 e 2018), resultando em dois novos conjuntos, com 1097 e 1184 entradas para os dados de 2017 e 2018 respectivamente, como mostra a figura 2.

Figura 2 – Visualização do arquivo CSV após tratamento.

	C	D	E
	Track Name	date	
laTOkWBF	Deu Onda	2017-01-01	
‹38njON	Hear Me Now	2017-01-01	
K0Qfg8L	10% - Ao Vivo	2017-01-01	
yD4WKf	Eu Sei de Cor - Ao Vivo | Acústico	2017-01-01	
!mVwAHll	Meu Coração Deu PT - Ao Vivo	2017-01-01	
VDzPAjs	Medo Bobo - Ao Vivo	2017-01-01	
xx4nK9	Sim ou não (participação especial Maluma)	2017-01-01	
2FVITVw	Closer	2017-01-01	
jmtU9ojo	50 Reais	2017-01-01	
C6npX95	Infiel - Ao Vivo	2017-01-01	
WpGK	Decide Aí - Na Praia / Ao Vivo	2017-01-01	
Qalq6uT	Te Assumi Pro Brasil - Ao Vivo	2017-01-01	
iYFLCW9	Malandramente	2017-01-01	
sjqml7p	Cold Water (feat. Justin Bieber & MØ)	2017-01-01	
ltKl66Z	Starboy	2017-01-01	
wHKlps8	O Nosso Santo Bateu - Na Praia / Ao Vivo	2017-01-01	
J3Fk945m	Let Me Love You	2017-01-01	
GzQalKU	Eu, Você, O Mar e Ela	2017-01-01	
FhlP7WX	Como Faz Com Ela - Ao Vivo	2017-01-01	
J8dNsUE	E Essa Boca Aí? - Ao Vivo	2017-01-01	
:aRqKoK	Paredes - Ao Vivo	2017-01-01	
FCJtw	Como É Que a Gente Fica - Ao Vivo	2017-01-01	

Fonte: Elaborada pelo autor.

2.2.2 Coleta de indicadores econômicos

Com as informações das músicas já obtidas e tratadas, passou-se à pesquisa sobre dados e indicadores de economia. Foi decidido, pela simplicidade e disponibilidade das informações, utilizar as métricas: Taxa Selic ("Sistema Especial de Liquidação e de Custódia", representa a taxa de juros) (SECRETARIA DE TECNOLOGIA DA INFORMACAO, 2019b), variação do IPCA ("Índice Nacional de Preços ao Consumidor Amplo", índice inflacionário oficial) (IBGE, 2019c) e renda familiar média, disponível pelo PNAD, ou Pesquisa Nacional por Amostra de Domicílios (IBGE, 2019b). A pesquisa inicial pelos indicadores se deu pelo portal brasileiro de dados abertos oficial do governo (SECRETARIA DE TECNOLOGIA DA INFORMACAO, 2019a), por onde os dados foram obtidos diretamente (por *download* do arquivo) ou através do sistema referenciado no portal. No nosso caso, os dados sobre a taxa Selic foram obtidos pelo primeiro método, e os demais por meio do segundo. No caso, o sistema direcionou à base do Sistema IBGE de Recuperação Automática (SIDRA) (IBGE, 2019a), portal oficial que armazena dados provenientes das pesquisas do IBGE (Instituto Brasileiro de Geografia e

Estatística). Por meio dele, foram referenciadas as tabelas das pesquisas apropriadas, PNAD para a renda média e IPCA para o próprio indicador, e selecionou-se os índices e períodos pertinentes. No caso dos indicadores, foi adotado tratamento manual dos arquivos obtidos, visto que em sua maioria eram correções mais simples sem recorrer a um automatizador, e que compreendiam um número baixo de dados. No caso da tabela sobre a taxa Selic, não foi possível filtrar o intervalo de tempo que nos era interessante (2017-2018), sendo necessária a remoção das tuplas das datas fora da faixa de tempo. No caso das outras duas tabelas, o sistema SIDRA exportou os dados em um formato altamente estilizado, apresentando os dados de maneira que o interpretador do Pandas não era capaz de reconhecer e incluindo diversas outras informações irrelevantes ao contexto do algoritmo, como legendas. Assim, foi nescessário reorganizar os dados e remover as entradas decorativas das tabelas. Por fim, como a referência para a junção dos dados seria por data, os formatos das datas listadas nessas tabelas tiveram que ser retrabalhados, adaptando o formato brasileiro ("dia/mês/ano") usado nessas tabelas para o formato "ano-mês-dia", mais comum no contexto de tecnologia e usado nas listas de músicas, para facilitar a união dos dados.

2.2.3 Análise de recursos de áudio

Os dados coletados sobre as músicas até esse ponto não apresentavam relevância para o que esse projeto se propôs: isto é, não continham dados descritivos das qualidades musicais e de áudio das faixas para análise. Para tanto, valeu-se então da API oficial do *Spotify*. Dentre os recursos de acesso disponibilizados pelo serviço, é de maior interesse o recurso de *Audio Features*, ou recursos de áudio, das faixas. Esse *endpoint*, dados um ou mais códigos de identificação, nos retorna uma análise com várias características musicais das obras consultadas, como a escala usada na composição, o tempo ou número de pulsos por minuto que determinam o andamento e velocidade da música, etc.

Para obter essas informações para cada entrada de dados das nossas músicas foi escrito um *script Python* que importa os dois arquivos contendo as listas de músicas tratadas, realiza a união de suas tuplas, e para cada 50 entradas realiza uma chamada para a API do serviço, conforme o algoritmo 1:

Algoritmo 1 – Algoritmo de Importação de Recursos de Áudio

Entrada: Tabela de Músicas 2017 T1 = [id1,id2,...,idn] e Tabela de Músicas 2018 T2 = [id1,id2,...,idm].

```
1.
2.    Seja Tabela Geral T3 = T1 U T2;
3.    Para cada tuplas t em T3, faça
4.        Se t contém nulo, então
5.            Elimina t
```

6. └
7. **Seja** *Conjunto de recursos das músicas R = [];*
8. **Para cada** *50 tuplas t em T3,* **faça**
9. **Seja** *Lista de recursos de uma música vi = [propriedade1, propriedade2,...,propriedaden];*
10. **Importe** *Análise de áudio r = [v1,v2,...,v50]*
11. **Acrescente** r *ao final de* R*;*
12. └
13. **Exporte** R **como** *arquivo;*
14. └

Desse modo, gera-se um novo arquivo, contendo as informações das propriedades sonoras disponíveis para análise pelo *Spotify* de todas as músicas indexadas. As chamadas *http* para o *endpoint* relevante à obtenção dos dados foram realizadas com auxílio do pacote *Spotipy*, que fornece uma interface simplificada para programas *Python* interagirem com a API em questão.

2.2.4 União dos dados de áudio com os *sets* econômicos

A última etapa no tratamento dos *datasets* antes de poder passar à análise propriamente dita foi a união dos dados de índices econômicos com o recém gerado conjunto de atributos das faixas. Nesse ponto do projeto o desafio foi o de unir os arquivos de maneira a alinhar as datas corretamente. Isso porque, apesar de a taxa Selic e as faixas terem registros com precisão do dia, os dados de renda média e IPCA apenas registram variações mensais. Além disso, mesmo dentro do conjunto da tabela sobre a taxa Selic há várias entradas de dias faltantes, o que gera uma incorrespondência com as músicas desse período. Assim, o algoritmo deveria indexar os dados econômicos de 3 arquivos separados de acordo não apenas de acordo com a equivalência do dia da análise, mas também com a do mês dependendo da tabela sendo cruzada, além de considerar e tratar entradas nulas.

A primeira versão da rotina está descrita no algoritmo 2.

Algoritmo 2 – Algoritmo de Importação de Recursos de Áudio

Entrada: Tabela de recursos de áudio T1 (onde ti é uma tupla de dados) = [t1,t2,...,tn], Conjunto de tabelas de índices econômicos T2 (onde u1 é o conjunto de dados representando variação do IPCA, u2 a renda média e u3 a taxa Selic) = [u1,u2,u3]

1.
2. **Para cada** *tuplas t em T1,* **faça**
3. **Seja** *T3 = um conjunto vazio [];*
4. **Seja** *d1 = dia da análise do áudio, em formato "yyyy-mm-dd";*
5. **Seja** *d2 = primeiro dia do mês da análise do áudio, em formato "yyyy-mm-dd";*

```
6.          Para cada tabela u em T2, faça
7.              Para cada tupla v em u;
8.                  Se u['data'] = d1 ou u['data'] = d2
9.                      Acrescente u ao final de T3;
10.
11.
12.
13.         Para cada tuplas x em T3
14.             Seja i = Índice de iteração de T3;
15.             Se x Contém nulo
16.                 Seja c = coluna da célula de valor nulo;
17.                 Seja f = próximo valor não nulo em c;
18.                 Seja b = último valor não nulo em c;
19.                 Se existe T3[i+1]
20.                     Atribua x[c] = f;
21.                     Senão, atribua x[c] = b
22.
23.
24.         Acrescente a primeira tupla x1 de T3 a t;
25.
26.     Exporte T1 como arquivo;
27.
```

No entanto, o mesmo apresentou problemas severos de performance (como pode ser deduzido pelo npumero de iterações aninhadas) e resultados inconsistentes, principalmente devido a problemas com o algoritmo de concatenação do *Pandas* para unir os resultados de pesquisa. Assim, foi trabalhada uma nova versão do programa, rodando um algoritmo com menos código e que, usando mais recursos prontos de otimização do *Python* e *Pandas*, conseguiu fazer a mescla dos indicadores econônicos às informações de áudio das faixas de modo mais performático e com melhor consistência.

Código 1 – Script de mesclagem entre os dados da análise de áudio e indicadores econômicos

```
import config
from fycharts import SpotifyCharts
import pandas as pd
import numpy as np
import glob
from datetime import datetime as dt

def getMonthBegin(date):
```

```
        monthBegin = dt.strptime(date, '%Y-%m-%d')
        monthBegin = dt(monthBegin.year, monthBegin.month,1)
        return dt.strftime(monthBegin, '%Y-%m-%d')

def matchValueByDate(date1,date2,df,col):

        matchId = np.where(date1.values == date2)[0]
        if matchId.size == 0 and col != 'daily_rate':
                date2 = getMonthBegin(date2)
                matchId = np.where(date1.values == date2)[0]
                if matchId.size == 0:
                        return None

        if col == 'daily_rate' and matchId.size == 0:
                return None
        else:
                val = df.loc[matchId[0],col]
        return val

def generateTableset(country):
        audioFeats = pd.read_csv(config.DATA['output_path']+'/'+country+'
        econTables = fetchEconomicData(country)
        ecoData = []

        for df in econTables:
                cols = list(df)
                for col in cols:
                        if(col == 'date'):
                                continue
                        audioFeats[col] = audioFeats.apply(lambda row: m

        print("NEW SET: \n {}".format(audioFeats));
        audioFeats.to_csv(config.DATA['output_path']+'/'+country+'/audio-

def fetchEconomicData(country):
        econFiles = glob.glob(config.DATA['data_path']+'/economics/'+cour
        econTables = []
        for file in econFiles:
                df = pd.read_csv(file,index_col=[0])
```

```
                econTables.append(df)
    return econTables
```

Apesar de, à primeira vista, haver indicadores de que a performance seria similiar entre ambos, como são usados recursos mais otimizados do *Pandas* para as iterarações na tabela de parâmetros sonoros e não serem realizadas tantas operações de concatenação entre tuplas e tabelas já reduziu a maior parte do tempo de execução inicial. Após averiguação da tabela gerada e algumas correções no algoritmo, passou-se à análise propriamente dita usando o arquivo gerado.

2.3 Algoritmo de análise

Paralelamente à criação dos *scripts* para gerar e tratar as tabelas de dados foi construído o conjunto de algoritmos para pré-processar e executar a análise dos dados. Até que os *datasets* tratados ficassem prontos, para fins de teste do *software* foram usados conjuntos prontos (como os vindos do pacote *scikit-learn*, ou de outras fontes como o Kaggle), ou partes das tabelas já obtidas (como a tabela de análise de áudio assim que o *crawler* ficou pronto). O trabalho dividiu-se primariamente em organizar um pré-processador, a execução propriamente dita da análise (treinamento e teste do algoritmo) e retorno dos resultados.

2.3.1 Pré processador

O objetivo do pré-processador foi o de encontrar os melhores parâmetros para desempenho do regressor e preparar os dados para a análise final. Dentro da preparação dos dados, o primeiro passo foi eliminar as entradas nulas dos dados de análise e do conjunto alvo a ser previsto. Apesar de esse tratamento já ter sido aplicado em diversas etapas do processamento e agrupamento dos dados, a tabela final ainda não havia sido tratada, sendo necessário passar a esse passo, já que entradas nulas poderiam, na melhor das hipóteses, viciar ou interferir nos resultados, e na pior, causar erros no algoritmo, forçando sua parada.

Passada essa primeira etapa, foram removidas as colunas inconsequentes ou danosas à análise. Como o objetivo era uma predição de características musicais baseada em índices econômicos, dados como o identificador e data de coleta podem e devem ser ignorados para não interferirem de alguma forma nos resultados. Da mesma forma, deve-se eliminar o conjunto a ser calculado da lista de índices econômicos, colocando-o em uma variável separada que, para fins de clarificação, será referida daqui para frente como *target*. Isso porque, do contrário, corre-se o risco de viciar o preditor a usar a própria entrada como parâmetro de previsão, nos dando resultados falsamente precisos. O *target*, e consequentemente as colunas a serem separadas, foi selecionado de maneira diferente para experimentação, de modo que o processo estará descrito com maior clareza nessa seção do trabalho.

Por fim, passou-se ao seletor de parâmetros. O método usado para o treino do modelo e predição dos atributos das faixas, o SVR (*Support Vector Regressor*) já veio integrado à biblioteca *scikit-learn*, mas apesar dessa conveniência ainda era necessário ajustar o conjunto de parâmetros usado pela função, bem como o *Scaler* usado. Para o algoritmo usado, os parâmetros para ajuste mais interessantes são: *kernel* (algoritmo de manipulação dos dados para uso na SVM), *C* (parâmetro de punição de erros, para controle de rigidez nos resultados), e *gamma* (parâmetro de controle de viés dos dados). O processo de seleção se deu como descrito pelo algoritmo 3:

Algoritmo 3 – Algoritmo de Seleção de Parâmetros para SVR

1.
2. ***Seja*** *M o modelo de regressão (SVR)*
3. ***Seja*** *lista de Kernels padrão do scikit-learn: K = ['rbf','linear','poly','sigmoid'];*
4. ***Seja*** *lista de valores para parâmetro C: C = [0.001,0.01,0.1,1,10,100];*
5. ***Seja*** *lista de valores para parâmetro Gamma: G = [0.001, 0.01, 0.1, 1, 10, 100];*
6. ***Seja*** *n = o número de combinações a ser testado;*
7. ***Seja*** *H = conjunto de resultados do coeficiente de determinação r;*
8. ***Seja*** *Conjunto de parâmetros B = [];*
9. ***Para n*** *combinações* b *de* K, C *e* G, ***faça***
10. ***Treine*** *M com os parâmetros* b*;*
11. ***Seja*** *h = coeficiente de determinação r de* $M(b)$*;*
12. ***Acrescente*** *h a H;*
13. ***Se*** *h =* ***Máximo*** *em H*
14. $B = b$
15.
16.
17. ***Retorne*** *B;*
18.

Para realizar a parte de combinação de parâmetros e treinamento/teste do modelo foi inicialmente utilizada a classe *GridSearchCV* do *scikit-learn*. Entretanto, principalmente ao tentar operações com *targets* compostos por múltiplas colunas de dados, a performance da mesma caiu drasticamente. Optou-se por um pacote similar chamado *BayesSearchCV* de um projeto filho: o *scikit-optimize*, após verificar melhora no tempo de execução. Apesar de ainda ser um processo lento e oneroso, foi necessário para maximizar a precisão dos resultados. Um fluxo similar foi aplicado para os *scalers*, onde testou-se apenas dois dos mais comuns: o *StandardScaler* e o *MinMaxScaler*. São treinados dois modelos usando o mesmo conjunto de

dados e método de análise e retorna-se o mais bem sucedido de acordo com o coeficiente de determinação (R^2) 2.

Apesar da carga de processamento e tempo gasto, os procedimentos acima foram aplicados diversas vezes, para ajustar o modelo a diferentes conjuntos de *targets* e, por consequente, de dados de análise.

2.3.2 Visualização dos resultados

Por fim, foi escrito um conjunto pequeno de rotinas para exibição das métricas de desempenho dos modelos gerados. Os indicadores escolhidos foram: indicador de variação explicada, erro médio absoluto, erro quadrado médio, erro médio quadrado logarítmico, desvio mediano absoluto e coeficiente de determinação, ou R^2 2. As funções simplesmente exibem o valor de cada um no terminal baseado na predição de um modelo determinado. Apesar de ser mais "cru", foi um indicador visual suficiente para avaliar o desempenho do algoritmo para diferentes parâmetros, *sets* de dados e *targets*.

2.4 Experimentação e resultados

Por fim passou-se à fase de experimentação. O objetivo dessa etapa foi testar diversas combinações de *targets* para determinar o modelo com previsão mais precisa e, portanto, quais os conjuntos características de áudio o algoritmo melhor consegue prever. Foi escrito um *script* simples para acessar o pré processador e, com o modelo retornado, executar o treino e teste, exibindo posteriormente os desvios descritos acima.

As características das músicas a serem estimadas e suas descrições são como detalhado pela tabela 1, montada a partir da documentação oficial da API do *Spotify*.

Tabela 1 – Descrição das características da análise de áudio

Nome do campo	**Descrição**
duration_ms	Duração da faixa em milissegundos(ms).
key	Tom estimado em que a faixa foi composta. É mapeada por números inteiros de acordo com notação numérica americana, onde 0 equivale à nota dó, 1 equivale a dó sustenido/ré bemol, e assim por diante até a nota si, que equivale ao numeral 11. Caso não haja estimativa do tom, é usado o valor -1.
mode	Indicador da modalidade (maior ou menor) predominante na melodia da faixa. O número 1 representa modalidade maior e o 0, menor.

Tabela 1 – Descrição das características da análise de áudio

Nome do campo	**Descrição**
time_signature	A notação de tempo estimada como predominante da faixa. É uma convenção do número de pulsos (batidas) em um ciclo da música (compasso).
acousticness	Um gradiente de confiança de 0 a 1 do quão "acústica"(isto é, o quão ausentes são elementos eletrônicos como: distorções, *MIDI*, etc. na sua composição, gravação ou produção) é uma faixa, onde 1 representa alta confiança da faixa ser "acústica"e 0, baixa confiança.
danceability	Gradiente de 0 a 1 descrevendo o quão adequada a música é para dançar, baseado em uma combinação de elementos da música como: tempo, estabilidade rítmica, força do pulso (ou batida), e regularidade em geral. O número 0 representa baixa adequação à dança, e o 1, alta.
energy	Medidor de 0 a 1 que representa a medida de intensidade e atividade percebidas na música. Tipicamente, faixas com maior "energia"são mais rápidas e barulhentas e são atribuídas um valor mais próximo de 1, enquanto a faixas "calmas"atribui-se um índice mais próximo de 0. Fatores perceptuais que afetam esse atributo incluem: variedade na dinâmica (ou intensidade sonora) da peça, sonoridade percebida, timbre, toada inicial e entropia em geral.
instrumentalness	Preditor se uma música contém ou não elementos vocais, isto é, com uso de voz humana (considera-se apenas a dicção clara de palavras, fonemas simples e isolados são desconsiderados). Quanto mais próximo o valor é de 1, maior a probabilidade de a obra não conter conteúdo de voz humana. Valores acima de 0.5 pretendem representar obras instrumentais, mas índices mais próximos de 1 inferem maior confiança na medição.
liveness	Detecta a presença de uma audiência "ao vivo"na gravação, representado em um gradiente de 0 a 1. Valores mais próximos de 1 representam maior probabilidade de uma faixa ter sido gravada com um público espectador.
loudness	A sonoridade geral de uma faixa, medida em decibeis (dB). Os valores constituem uma média da intensidade sonora de toda a gravação, e é usada para comparar a sonoridade relativa entre músicas. Os valores típicos se encontram entre -60dB e 0dB

Tabela 1 – Descrição das características da análise de áudio

Nome do campo	**Descrição**
speechiness	Intervalo de 0 a 1 que representa a confiança de uma faixa apresentar elementos de palavra falada, isto é, de não ser apenas instrumental. Quanto menos elementos musicais além da voz, mais próximo de 1 e valores acima de 0.66 tendem a representar faixas exclusivamente com texto declamado, sem demais elementos musicais ou instrumentos. Valores entre 0.33 e 0.66 representam em geral faixas híbridas e abaixo de 0.33, música ou outros formatos não discursivos.
valence	Uma métrica de 0 a 1 descrevendo a "positividade"da música, com maior valência representando músicas mais "positivas"(ex.: eufóricas, alegres, animadas) e menor, mais "negativas"(ex.: triste, deprimente, irada).
tempo	O tempo médio estimado de uma faixa, em batidas por minuto(bpm). Na terminologia musical, o tempo de uma composição representa sua velocidade ou andamento e deriva diretamente da duração média da batida.

Para avaliação a performance do preditor foi utilizado um conjunto de métricas de erro próprias para algoritmos de regressão. Sua descrição se encontra na tabela 2.

Tabela 2 – Métricas de avaliação do algoritmo de regressão.

Métrica	**Descrição**
Indicador de variação explicada	Mede a proporção em que o modelo considera a dispersão (ou variância) de um conjunto de dados. Tenha que $\widehat{y}$ é o valor de saída alvo, y é o valor de saída real e Var é a variância (quadrado do desvio padrão), a variância explicada V é expressa por: $V(y, \widehat{y}) = 1 - \frac{Var\{y - \widehat{y}\}}{Var\{y\}}$. Melhor valor possível é 1.
Erro médio absoluto	Valor médio absoluto esperado para o erro. Tenha que $\widehat{y}_i$ é o valor previsto da i-ésima amostra e y_i é o valor real observado, o erro (EMA) estimado sobre n amostras é expresso por: $EMA(y, \widehat{y}) = \frac{1}{n}\sum_{i=0}^{n-1} \lvert y_i - \widehat{y}_i \rvert$. Melhor valor possível é 0. Não admite valores negativos.

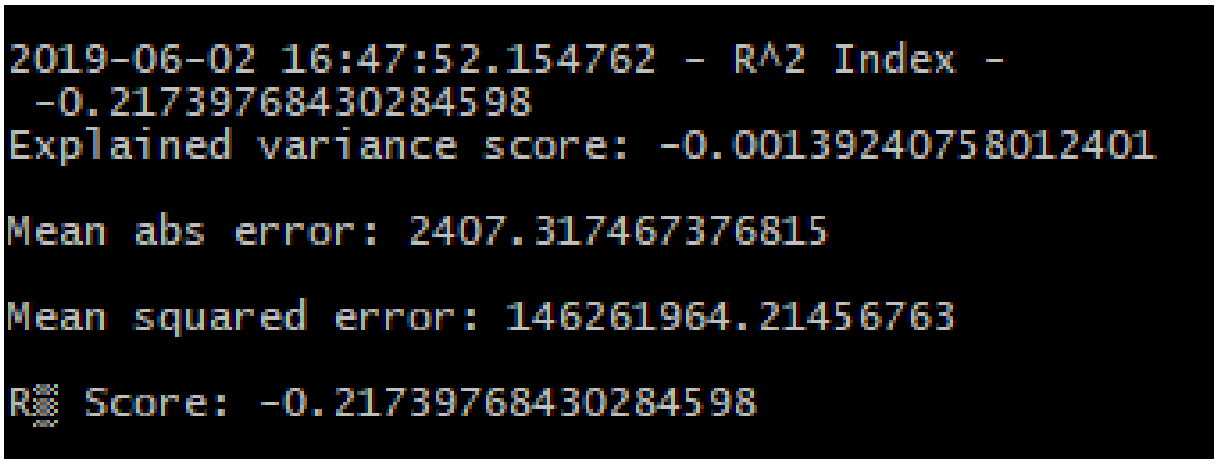
Tabela 2 – Métricas de avaliação do algoritmo de regressão.

Métrica	**Descrição**
Erro quadrado médio	Valor médio esperado para o erro quadrático. Atribui maior peso a erros maiores e menor peso a erros menores. Tenha que $\widehat{y}_i$ é o valor previsto da i-ésima amostra e y_i é o valor real observado, o erro (EMQ) estimado sobre n amostras é expresso por: $EMQ(y, \widehat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \widehat{y}_i)^2$. Melhor valor possível é 0. Não admite valores negativos.
Erro quadrado médio logarítmico	Valor esperado para o quadrado da transformação logarítmica do erro. Atribui maior peso a erros maiores e menor peso a erros menores. Tenha que $\widehat{y}_i$ é o valor previsto da i-ésima amostra e y_i é o valor real observado, o erro (EQML) estimado sobre n amostras é expresso por: $EQML(y, \widehat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}(\log_e(y_i + 1) - \log_e(\widehat{y}_i + 1))^2$, onde $\log_e x$ é o logaritmo natural de x. essa métrica é melhor adequada para modelos de crescimento exponencial. Como nota de atenção, ela também penaliza subestimativas mais do que sobrestimativas nas previsões. Melhor valor possível é 0. Não admite valores negativos.
R^2	Também chamado coeficiente de determinação. Representa a proporção da variância explicada pelas variáveis independentes do modelo. Provê um incador da adequação e, assim, a medida do quão bem amostras ainda não analisadas serão previstas pelo modelo por meio da proporção da variância explicada. Tenha que $\widehat{y}_i$ é o valor previsto da i-ésima amostra, y_i é o valor real observado, $\overline{y} = \frac{1}{n}\sum_{i=1}^{n}\widehat{y}_i$ e $\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2 = \epsilon_i^2$, o R^2 estimado sobre n amostras é expresso por: $R^2(y, \widehat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$. Melhor valor possível é 1 e pode retornar valores negativos. Um modelo constante que sempre prediz o valor esperado de y, independente das características de entrada possui valor 0.

Fonte: (SCIKIT-LEARN DEVELOPERS, 2019)

O primeiro teste foi sobre o conjunto de todas as características de áudio como *target*, isto é, o modelo tenta prever todos os parâmetros baseado nos indicadores econômicos. Os resultados foram como descrito na figura 3 e tabela 3.

Figura 3 – Resultados de predição de todos os atributos de áudio.



Fonte: Elaborada pelo autor.

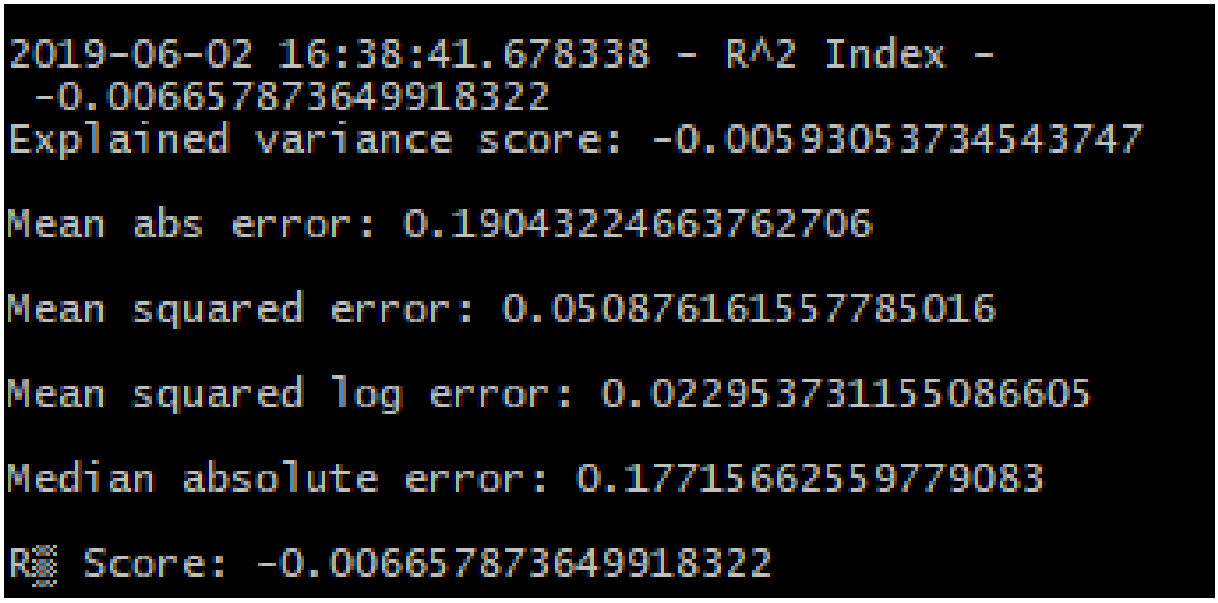
Tabela 3 – Resultados de predição de todos os atributos de áudio.

Métrica	**Resultado**
Indicador de variação explicada	≃ -0,0014
Erro médio absoluto	≃ 2407,3175
Erro quadrado médio	≃ 146261964,2146
R	≃ -0,2174

Fonte: Elaborada pelo autor.

Como observado, o erro calculado para todas as métricas foi grande, retornando assim resultados insatisfatórios. Uma das possíveis razões foi a inadequação do algoritmo de regressão escolhido para operar com o número de variáveis (treze no total) a ser previsto. Experimentou-se então o extremo oposto, construindo um *target* com apenas uma variável. A variável escolhida foi a *valence*, ou valência. A escolha do parâmetro se deu por duas razões: sua ligação direta com a qualidade das emoções percebidas; seu cálculo, que por ser feito a partir de diversos outros atributos, abre a possibilidade de que eles sejam adequadamente representados por esta única variável. Os novos resultados estão descritos na figura 4 e tabela 4.

Figura 4 – Resultados de predição da valência.



Fonte: Elaborada pelo autor.

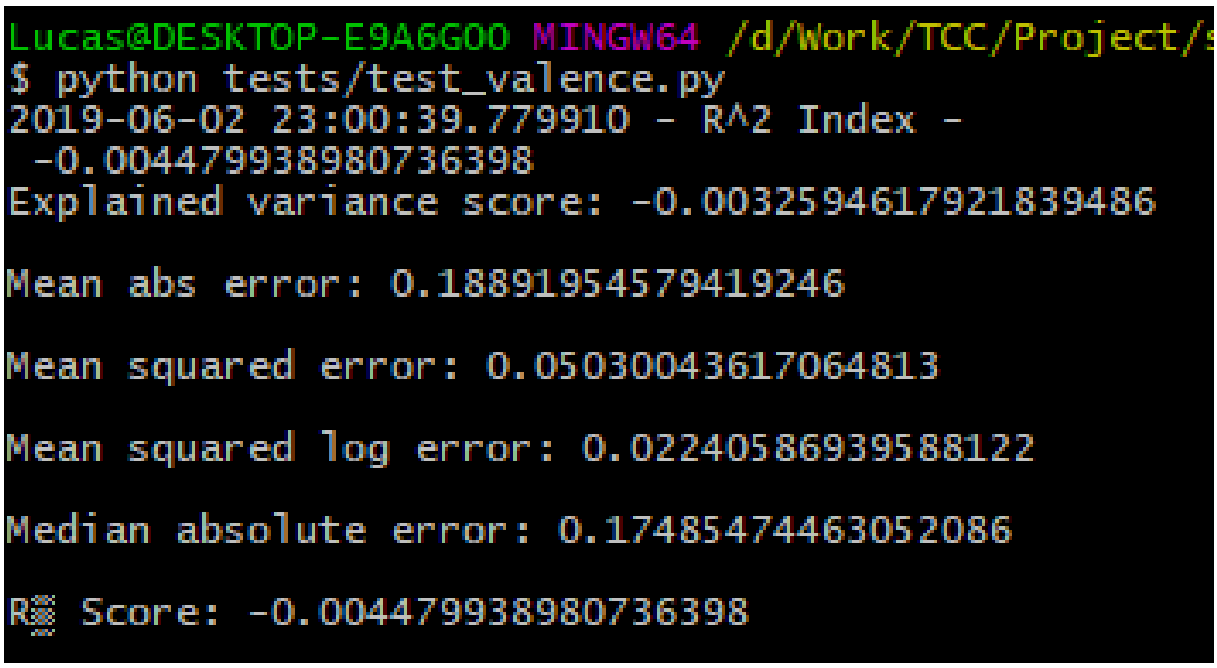
Métrica	Resultado
Indicador de variação explicada	≃ -0,0059
Erro médio absoluto	≃ 0,1904
Erro quadrado médio	≃ 0,0509
Erro quadrado médio logarítmico	≃ 0,0229
R	≃ -0,0067

Tabela 4 – Resultados de predição da valência.

Fonte: Elaborada pelo autor.

Apesar dos parâmetros R e de variação explicada exporem que o modelo não é o mais adequado, os resultados dos demais erros foram satisfatórios, com erro médio baixo em todos as medições. No entanto, por ser um valor calculado discretamente pelo serviço do *Spotify*, não há prova empírica de que a valência representa os demais atributos omitidos. Assim, foi calculado um último teste, onde analisou-se apenas a tabela de recursos de áudio e tentou-se predizer a valência a partir das demais qualidades. Os resultados estão descritos na figura 5 e tabela 5.

Figura 5 – Resultados de predição da valência baseado nos demais atributos das faixas.



Fonte: Elaborada pelo autor.

Métrica	Resultado
Indicador de variação explicada	≃ -0,0033
Erro médio absoluto	≃ 0,1889
Erro quadrado médio	≃ 0,0503
Erro quadrado médio logarítmico	≃ 0,0241
R	≃ -0,0045

Tabela 5 – Resultados de predição da valência.

Fonte: Elaborada pelo autor.

Observa-se que os resultados foram similares aos obtidos acima. Apesar da baixa variação, que nos indica uma boa predição, infelizmente novamente constata-se pelos indicadores R e de variação explicada que o modelo não é o mais adequado.

3 Conclusão

Conforme mencionado, apesar de alguns dos resultados parecerem promissores, não podemos com segurança confirmar uma correlação entre os dados econômicos e as características das faixas extraídos. Como pontos de consideração sobre como melhorar o desempenho em geral podemos destacar, a partir do próprio relatório de erros dos experimentos, a inadequação do modelo, que muito provavelmente advém do uso de um algoritmo sub-ótimo para o problema em questão. Apesar do poder do método de SVM para classificação e da possibilidade de seu uso para regressão (como fizemos), ficou em dúvida se outros algoritmos de regressão, especialmente os especializados em tratar múltipla variáveis, não seriam mais adequados e, por consequente, retornariam resultados mais sólidos. Da mesma forma, apesar do cuidado na manipulação e extração dos dados, não se pode excluir a hipótese de viés nas amostras ou de escolha imprecisa das variáveis, tanto as econômicas quanto as pertinentes às qualidades das músicas. Também pode-se oferecer o argumento que, pesquisando o contexto de outras regiões que não apenas o Brasil poderiam se obter resultados mais conclusivos, já que o número de dados e contextos diferentes minimizariam, em tese, a influência de outros possíveis fatores nos resultados.

Referências

ALJANAKI, A.; YANG, Y.-H.; SOLEYMANI, M. Emotion in music task at mediaeval 2015. In: *MediaEval*. [S.l.: s.n.], 2015. Citado na página 10.

GITHUB. *Spotify Charts API*. 2019. Disponível em: <https://github.com/kelvingakuo-/fycharts>. Acesso em: 14 de maio, 2019. Citado na página 13.

GITHUB. *Spotify Charts API*. 2019. Disponível em: <https://github.com/luckponte-/fycharts>. Acesso em: 14 de maio, 2019. Citado na página 14.

IBGE. *SIDRA*. 2019. Disponível em: <https://sidra.ibge.gov.br/>. Acesso em: 22 de maio 2019 2019-05-22. Citado na página 15.

IBGE. *SIDRA Pesquisa Nacional por Amostra de Domicílios Contínua mensal*. 2019. Disponível em: <https://sidra.ibge.gov.br/Tabela/6387>. Acesso em: 22 de maio 2019 2019-05-22. Citado na página 15.

IBGE. *SIDRA Índice Nacional de Preços ao Consumidor Amplo*. 2019. Disponível em: <https://sidra.ibge.gov.br/tabela/1419>. Acesso em: 22 de maio 2019 2019-05-22. Citado na página 15.

KAUSAR, M. A.; DHAKA, V. S.; SINGH, S. K. Web crawler: A review. *International Journal of Computer Applications*, v. 63, p. 31–36, 02 2013. Citado na página 13.

KRUMHANSL, C. L. Music: A link between cognition and emotion. *Current Directions in Psychological Science*, v. 11, n. 2, p. 45–50, 2002. Disponível em: <https://doi.org/10.1111-/1467-8721.00165>. Citado na página 10.

LAMERE, P. *Spotipy Documentation*. 2019. Disponível em: <https://spotipy.readthedocs-.io>. Acesso em: 23 de fevereiro, 2019. Citado na página 13.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado na página 13.

SCIKIT-LEARN DEVELOPERS. *Scikit-learn Regression metrics*. 2019. Disponível em: <https://scikit-learn.org/stable/modules/model_evaluation.htmlregression-metrics>. Acesso em: 20 de março, 2019. Citado na página 25.

SECRETARIA DE TECNOLOGIA DA INFORMACAO. *Portal Brasileiro de Dados Abertos*. 2019. Disponível em: <http://dados.gov.br/>. Acesso em: 22 de maio 2019 2019-05-22. Citado na página 15.

SECRETARIA DE TECNOLOGIA DA INFORMACAO. *Taxa de juros - Selic*. 2019. Disponível em: <http://dados.gov.br/dataset/11-taxa-de-juros-selic>. Acesso em: 22 de maio 2019 2019-05-22. Citado na página 15.

SINGH, S.; SINGH, P.; GARG, R.; MISHRA, P. K. Big data: Technologies, trends and applications. v. 6, p. 4633–4639, 10 2015. Citado na página 10.

SPOTIFY AB. *Spotify API Documentation*. 2019. Disponível em: <https://developer-.spotify.com/documentation/web-api/>. Acesso em: 22 de março, 2019. Citado na página 14.

SPOTIFY AB. *Spotify Charts*. 2019. Disponível em: <https://spotifycharts.com>. Acesso em: 25 de abril, 2019. Citado 2 vezes nas páginas 13 e 14.