



**OUVIDO MUSICAL: EXPLORANDO METADADOS EM UM SISTEMA DE
RECOMENDAÇÃO DE MÚSICAS**

Por

DIEGO CORRÊA DA SILVA

Trabalho de Graduação



Universidade Federal da Bahia
dcc@ufba.br
wiki.dcc.ufba.br/DCC/

SALVADOR/2018



Universidade Federal da Bahia
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

DIEGO CORRÊA DA SILVA

"OUVIDO MUSICAL: EXPLORANDO METADADOS EM UM SISTEMA DE
RECOMENDAÇÃO DE MÚSICAS"

Trabalho apresentado ao Programa de Bacharelado em Ciência da Computação do Departamento de Ciência da Computação da Universidade Federal da Bahia como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: *Frederico Araújo Durão*

SALVADOR, 2018

Diego Corrêa da Silva

Ouvido Musical: Explorando metadados em um sistema de recomendação de músicas/
Diego Corrêa da Silva. – SALVADOR, 2018-
83 p. : il. (algumas color.) ; 30 cm.

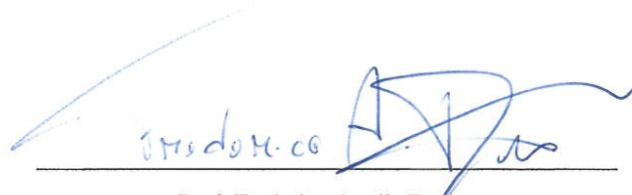
Orientador :Frederico Araújo Durão

Trabalho de Graduação – Universidade Federal da Bahia
Instituto de Matemática e Estatística
Departamento de Ciência da Computação, 2018.

1. Sistemas de Recomendação. 2. Turismo. I. Frederico Araújo Durão. II. Universidade Federal da Bahia. III. Instituto de Matemática e Estatística. IV. Departamento de Ciência da Computação. V. Ouvido Musical: Explorando metadados em um sistema de recomendação de músicas

CDU 02:141:005.7

Monografia apresentada por **Diego Corrêa da Silva** ao programa de Bacharelado em Ciência da Computação do Departamento de Ciência da Computação da Universidade Federal da Bahia, sob o título **Ouvido Musical: Explorando metadados em um sistema de recomendação de músicas**, orientada pelo **Prof. Frederico Araújo Durão** e aprovada pela banca examinadora formada pelos professores:



Prof. Frederico Araújo Durão
Departamento de Ciência da Computação/UFBA



Prof. Maurício Pamplona Segundo
Departamento de Ciência da Computação/UFBA



Prof. Rodrigo Rocha Gomes e Souza
Departamento de Ciência da Computação/UFBA

Dedico esta monografia à minha mãe que sempre me permitiu meios para estudar, amigos e professores que me deram todo o apoio necessário para chegar até aqui.

Pra desistir é muito fácil. Joga a bandeira e corre pro abraço, vai. O Monstro é grande sim, mas não é invencível. Pois este império é artigo perecível. Independente disso faça o que tu acredita, pois o mundo fica mudo a cada alma que abdica.

—B. NEGÃO

Resumo

A música é uma das mais antigas obras humanas. Passada de geração em geração e usada em diversos setores e com diversas finalidades. A evolução da música digital possibilitou, a partir dos anos 2000, o surgimento dos serviços de transmissão de música, o que produziu um efeito altamente lucrativo ao setor musical. Estes serviços se utilizam de sistemas de recomendação para encontrar faixas que venham a agradar aos usuários, coletando assim ricas informações a respeito do que ouvem. O volume de dados obtido possibilita um cenário rico em características que descrevem a ambos, fazendo, assim, com que as recomendações possam ser mais confiáveis. Sendo assim, é possível um cenário com escassez de dados onde se encontrem recomendações para seus usuários? O problema que este trabalho visa resolver é a dificuldade que os sistemas de recomendação tem em sugerir novas obras para os usuários diante de um cenário com escassez de informação, explorando combinações de metadados para encontrar recomendações. Ao longo deste trabalho um levantamento teórico será realizado, assim como um levantamento das aplicações comerciais, o sistema proposto que usará similaridade entre os metadados para encontrar as recomendações personalizadas. Os modelos necessários para o desenvolvimento deste sistema são apresentados e exemplificados. Os resultados mostram que é possível realizar recomendações em um cenário com escassez informação. As métricas usadas para avaliar o sistema foram *Mean Average Precision* e *Mean Reciprocal Rank*. A avaliação mostrou que em um ambiente desbalanceado onde apenas 11% das músicas são consideradas relevantes é possível obter, na métrica *MAP*, aproximadamente 20% de assertividade ao longo das listas de recomendações, assim como uma lista de recomendação com um número maior de faixas ajuda na obtenção de um melhor resultado. A avaliação também mostrou que é possível obter na métrica *MRR* mais de 35% de assertividade, assim como a lista de recomendação com um número maior de músicas interfere diretamente no resultado positivo da métrica.

Palavras-chave: Sistema de Recomendação, Músicas, Metadados, Lista de Recomendação, Filtragem Baseada em Conteúdo

Abstract

Music is one of the oldest humanity works. Passed from generation to generation and used in so many sectors and for many purposes. The evolution of digital music made possible, from the 2000s, onwards the emergence of music streaming services, that produced a highly lucrative effect on the music sector. These services use recommendation systems to find tracks that will appeals to users, collecting rich information about them and what they listen to. The obtained volume of data enables a scenario rich in characteristics that describe both, thus making recommendations more reliable. So, is possible to find recommendations in a scenario with scarcity of data? The problem this work aims to solve is the difficulty of recommendations systems have to suggesting new songs for users in front of a scenario with scarcity information, exploring metadata combinations to find recommendations. Throughout this work a theoretical survey will be conducted, as well as a survey of commercial applications, the proposed system, will use similarity between metadatas to find personalized recommendations. The necessary models for development this system are presented and exemplified. The results show it is possible to make recommendations in a scenario with scarcity information. The used metrics to evaluate the system results were Mean Average Precision and Mean Reciprocal Rank. The evaluation showed in an unbalanced environment where only 11% of songs are considered relevant is possible to obtain, in the metric MAP, approximately 20% of assertiveness along the recommendations lists, as well as a recommendation list with a larger number of songs helps to obtaining a better result. The evaluation also showed it is possible to obtain more than 35% assertiveness in the metric MRR, just as the recommendation list with a larger number of songs interferes directly with the positive metric result.

Keywords: Recommender Systems, Song, Music, Metadata, Recommendation List, Content-based Filtering

Sumário

1	Introdução	23
1.1	Motivação	24
1.2	Problema	25
1.3	Objetivo	25
1.4	Estrutura	26
2	Sistemas de Recomendação	27
2.1	Histórico	27
2.2	Conceitos	29
2.3	Tarefas realizadas por um Sistema de Recomendação	29
2.4	Técnicas de Recomendação	31
2.4.1	Filtragem Colaborativa	31
2.4.2	Filtragem Baseada em Conteúdo	32
2.4.3	Arquitetura de Sistemas de Recomendação Baseados em Conteúdo	33
2.4.4	Prós e Contras das Técnicas de Recomendação	35
2.5	Sumário	36
3	Música Digital	37
3.1	Introdução	37
3.2	Aplicações com Sistemas de Recomendação de Música	38
3.2.1	E-Commerce	38
3.2.2	Music Streaming	39
3.3	Sistema de Recomendação de Música	40
3.3.1	Filtragem Colaborativa	40
3.3.2	Filtragem Baseado em Conteúdo	42
3.3.3	Híbrida	42
3.3.4	Sensível ao Contexto	43
3.3.5	Marcação Social	43
3.3.6	Redes de Músicas	44
3.3.7	Visualização	44
3.3.8	Gerador de Lista de Reprodução	44
3.3.9	Recomendação em Grupo	45
3.4	APIs e Datasets	45
3.4.1	Dataset	45
3.4.1.1	Million Song Dataset	45

3.4.1.2	UCI	46
3.4.1.3	MusicBrainz	46
3.4.2	APIs	46
3.4.2.1	Last.fm	46
3.4.2.2	Spotify	47
3.5	Sumário	47
4	Ouvido Musical: Um Sistema de Recomendação de Música	49
4.1	Algoritmos	49
4.1.1	Medidas de similaridade baseada em conteúdo	49
4.1.2	TF-IDF	50
4.1.3	Similaridade do Cossenos	50
4.2	Modelagem dos dados	51
4.2.1	Banco de dados	51
4.2.2	Usuário	52
4.2.3	Item de Recomendação: Música	54
4.2.4	Modelo do usuário	54
4.2.5	Modelo da Música	55
4.2.6	Cálculo médio do interesse dos usuários sobre uma música	56
4.2.7	Modelo de Recomendação	57
4.3	Etapas de Execução	58
4.4	Tecnologias	59
4.4.1	Python	59
4.4.2	PostgreSQL	60
4.5	Sumário	61
5	Avaliação	63
5.1	Metodologia	63
5.2	Conjunto de dados	64
5.3	Métricas de Avaliação	65
5.3.1	Mean Average Precision - MAP	65
5.3.2	Mean Reciprocal Rank - MRR	67
5.4	Resultados	67
5.4.1	MAP	67
5.4.2	MRR	70
5.5	Discussão	72
5.6	Pontos de Melhorias	72
5.7	Sumário	73

6 Conclusão	75
6.1 Contribuições do Trabalho	76
6.2 Trabalhos Futuros	76
6.3 Sumário	77
Referências	78
Bibliografia	79

Lista de Figuras

2.1	Exemplificação do método de Filtragem Colaborativa.	32
2.2	Exemplificação do método de Filtragem Baseada em Conteúdo.	33
2.3	Arquitetura de alto nível de um Sistema de Recomendação Baseado em Conteúdo (Lops <i>et al.</i> , 2011).	34
3.1	Recomendações de álbuns no comércio eletrônico da <i>Amazon</i>	39
3.2	Recomendações de álbuns musicais no comércio eletrônico Submarino.	39
3.3	Lista de recomendações semanal do <i>Spotify</i> com 30 novas músicas para serem descobertas na semana.	40
3.4	Lista de recomendações de músicas do <i>Spotify</i> baseada no país do usuário.	41
3.5	Recomendações de estilos musicais do <i>Last.fm</i> baseada em <i>Tags</i>	41
3.6	Utilização do método de filtragem colaborativa baseado em item pelo <i>Last.fm</i>	41
3.7	Lista de recomendações do <i>Youtube</i> contendo músicas e trailers de filme.	42
4.1	Modelagem física do banco de dados do sistema, mostrando as tabelas e as colunas.	53
4.2	Conjunto das músicas composto por seus metadados.	55
4.3	Modelo do usuário com 4 músicas adicionadas na sua preferência.	55
4.4	Conjunto das Músicas, Modelo da Música e Modelo do Usuário.	56
4.5	Modelo de recomendação com as músicas similares entre D_u e C_u	58
4.6	Etapas de execução das atividades do sistema para gerar uma recomendação.	58
5.1	Histograma normalizado das músicas.	65
5.2	Histograma normalizado dos usuários.	66
5.3	MAP - Gráfico com linhas dos experimentos com 10000 músicas e três variações no número de usuários: 100, 500 e 1000.	68
5.4	MAP - Gráfico com caixas dos experimentos com 10000 músicas e três variações no número de usuários: 100, 500 e 1000.	69
5.5	MAP - Gráfico com linhas dos experimentos com 1000 usuários e três variações no número de músicas: 3000, 6000 e 9000.	69
5.6	MAP - Gráfico com caixas dos experimentos com 1000 usuários e três variações no número de músicas: 3000, 6000 e 9000.	69
5.7	MRR - Gráfico com linhas dos experimentos com 10000 músicas e três variações no número de usuários: 100, 500 e 1000.	71
5.8	MRR - Gráfico com caixas dos experimentos com 10000 músicas e três variações no número de usuários: 100, 500 e 1000.	71

5.9	MRR - Gráfico com linhas dos experimentos com 1000 usuários e três variações no número de músicas: 3000, 6000 e 9000.	71
5.10	MRR - Gráfico com caixas dos experimentos com 1000 usuários e três variações no número de músicas: 3000, 6000 e 9000.	72

Lista de Tabelas

4.1	Valores para o TF de cada documento.	51
4.2	Valores para o IDF de cada termo.	51
4.3	Valor balanceado do TF pelo IDF.	52
4.4	Matriz de similaridade música por música.	52
4.5	Metadado $\langle UID \rangle$ representando 5 usuários, cada usuário em uma linha. . . .	53
4.6	As músicas e seus metadados.	54
4.7	Modelos de 3 usuários diferentes.	56
4.8	Similaridade da música “ <i>Rock a My Soul</i> ” com o D_u sendo duas músicas similares. . . .	57
4.9	Similaridade da música “ <i>Easily</i> ” com o D_u sendo similar a uma música.	57
4.10	Lista encontrada pelo modelo de recomendação.	57
5.1	Informações sobre o conjunto de dados usado.	64
5.2	Exemplo do cálculo do MAP para tamanho 10 da lista.	67
5.3	Exemplo do cálculo do MRR para três usuários.	67

Lista de Acrônimos

APIs	Application Programming Interfaces	45
CD	Compact Disc	23
DVD	Digital Versatile Disc	23
HOSVD	Higher Order Singular Value Decomposition	43
MAP	Mean Average Precision	64
MFCCs	Mel-Frequency Cepstral Coefficients	42
MIDI	Musical Instrument Digital Interface	42
MP3	MPEG Audio Layer-3	23
MRR	Mean Reciprocal Rank	64
RR	Realimentação de Relevância	28
REST	Representational State Transfer	46

1

Introdução

Computadores fazem arte. Artistas fazem dinheiro.

—NAÇÃO ZUMBI (Da Lama ao Caos)

A música está presente em todas as civilizações existentes. Sendo usada em diversos setores e com diversas finalidades como: religiosas, propagandas, terapêuticas, amorosas, estúdiosas e outros. E com as descobertas das diversas formas de armazenar em meios digitais os sons acústicos. Facilitou a expansão do meio musical e assim dia após dia diversas destas músicas são criadas e lançadas por gravadoras ou por artistas independentes.

O surgimento da digitalização e tecnologias eficientes de compressão de áudio, como o MPEG Audio Layer-3 (MP3), realizou o sonho de tornar milhões de títulos de músicas acessíveis a milhões de usuários (Pachet, 2003). Como se apresentam em formatos digitais podem ser disponibilizadas em diversos meios como transmissão de música (*Music Streaming*) na Internet, Digital Versatile Disc (DVD), Compact Disc (CD) e outros. Em especial os serviços de transmissão de música na Internet tem uma vantagem sobre os outros meios (G1, 2017). Já que o usuário tem acesso a um acervo diversificado de músicas para ouvir, este que pode ser atualizado constantemente. Além de possibilitar o usuário mudar o gênero, o artista ou o álbum. Tudo isso sem precisar mudar a mídia física como no DVD e CD.

O surgimento desses serviços de transmissão de música produziu um efeito altamente lucrativo, impulsionando o crescimento do setor musical. No ano de 2016 empresas como Spotify¹, Apple Music², Youtube³ e Pandora⁴, geraram um aumento de 11,4% no faturamento da indústria musical, chegando a uma receita de 7.7 bilhões, que é a maior alta desde de 1998. Este crescimento vem aumentando ano após ano, em cinco anos estes serviços passaram de 9% para 51% do faturamento total da indústria musical (Estadão, 2017). Em 2017, os principais serviços de transmissão de música geraram 14,2 milhões de dólares por dia (News, 2018). Esses

¹<https://www.spotify.com/>

²<https://www.apple.com/br/music/>

³<https://www.youtube.com/>

⁴<https://www.pandora.com/>

serviços também permitem o armazenamento interno de diversos tipos de informação, como por exemplo: os gostos musicais dos usuários, músicas mais ouvidas, artistas mais ouvidos, músicas preferidas em países e/ou continentes, dando assim a oportunidade de constante expansão para área e satisfação para os usuários. No ano de 2015 os serviços supracitados ultrapassaram a marca de 1 trilhão de reproduções no ano (Forbes, 2015). O *Spotify* em 2015 teve 20 bilhões de horas de músicas reproduzidas, sendo mais de 20 mil faixas adicionadas diariamente, e em 2018 ultrapassou a quantidade de 30 milhões de músicas em seu catálogo (DMR, 2018). Não é difícil imaginar que num futuro próximo, tudo que for gravado estará disponível online e os serviços de música digital oferecerão bilhões de obras, com milhões a mais chegando a cada semana (Celma and Lamere, 2008).

1.1 Motivação

Com tantas músicas disponíveis para serem escolhidas, a busca por novidades envolve o usuário em um esforço enorme, sendo que muitas vezes o mesmo irá encontrar coisas que não o agrade, criando assim uma sobrecarga de informação musical (Hijikata *et al.*, 2006). É possível transferir este trabalho de encontrar trabalhos novos para um sistema automatizado, este que fique responsável por encontrar uma lista ranqueada de recomendações para cada usuário.

Os serviços de transmissão de música utilizam dos sistemas de recomendação para criar listas de músicas recomendadas das mais variadas formas, como por exemplo, as 50 mais ouvidas da semana ou do ano, músicas de rock dos anos 70, músicas de determinado país. Estas listas, porém, muitas vezes são geradas com um objetivo específico e não são personalizadas para cada usuário.

De acordo com Ricci *et al.* (2011), Sistemas de Recomendação são ferramentas de programas e técnicas que proveem sugestões de itens para os usuários. Estas sugestões são encontradas através de comparações entre itens e/ou usuários. Atualmente, sistemas de recomendação estão em diversos setores da Internet. Fields *et al.* (2011) define Sistemas de Recomendação como uma técnica ou método que apresenta a um usuário objetos sugeridos para consumo com base em seu comportamento passado.

Para realizar seu trabalho, um Sistema de Recomendação de Música necessita de dados sobre as preferências dos usuários e das características que descrevem as músicas. Aplicações como o *Last.fm* e o *Spotify* possui milhões de usuários e obras contendo, assim, um cenário rico de informações e possibilitando assim recomendações mais precisas. Na literatura de Sistemas de Recomendação é comum estudos que desejam enriquecer o cenário com dados provindos de diversas fontes (*Web*, redes sociais, contexto) para que assim as recomendações possam ser mais “úteis” aos usuários.

Entretanto o que acontece caso esses sistemas não possuam um cenário com riqueza de dados? Seria possível construir um sistema de recomendação que escolhesse músicas em um ambiente com escassez de informação? Quais abordagens esse sistema usaria? Como esses dados

podem ser utilizados para encontrar as recomendações? É possível utilizar alguma combinação de dos dados ou seria melhor trabalhar os atributos isolados?

1.2 Problema

O problema que este trabalho visa resolver é a dificuldade que os sistemas de recomendação têm em fazer sugestões para os usuários diante de um cenário com escassez de informação disponível, tendo em vista que a busca manual por novidades nem sempre é um trabalho agradável, requisitando do usuário tempo e concentração. A busca por uma nova música inicia-se a partir da peculiaridade musical do usuário, que escolhe uma informação para a busca de uma nova música ou pede recomendações a amigos e conhecidos, os quais irão recomendar músicas de suas preferências.

O desenvolvimento de um modelo de recomendação que identifique as trabalhos que são parecidos é apenas uma parte do problema. Já que identificar quais são as possíveis preferências do usuário requer uma interação com o sistema que o alimente, dando informações prévias sobre os seus gostos musicais. Logo, o problema conhecido como "partida a frio" (*cold-start problem*), que é um dos primeiros desafios de um sistema de recomendação, também é aplicável a um sistema de recomendação de música. Se o usuário não fornece nenhum tipo de informação o mesmo receberá recomendações de baixa qualidade até que esta interação forneça mais dados ao sistema para uma melhor seleção das opções.

1.3 Objetivo

Este trabalho propõe como solução a criação de um sistema, que em um cenário com escassez de informação, analisa a similaridade de metadados que descrevem as mídias como título, artista e álbum, utilizando alguma técnica de recomendação, que irá auxiliar a gerar melhores recomendações após cada interação do sistema com os dados do usuário. Sendo, estas recomendações, personalizadas para cada usuário e listadas para que o mesmo possa avaliá-las. Os objetivos específicos do trabalho são:

- Recuperar e processar os metadados das músicas;
- Definir o modelo do usuário de acordo com o que foi ouvido ou que tenha expressado interesse;
- Propor um modelo de recomendação utilizando algum algoritmo de similaridade;
- Avaliar a proposta com uma base de dados pública;

O problema da partida a frio não será abordado neste trabalho, visto que estamos interessados em trabalhar com os metadados da música. O sistema usará uma base de dados

disponibilizada pelo projeto *Million Songs Dataset* ⁵.

1.4 Estrutura

Neste Capítulo 1 motivamos e introduzimos o problema que este trabalho propõe resolver. Os próximos capítulos estão organizados na seguinte sequência: O Capítulo 2, que apresenta a base teórica a ser utilizada neste trabalho referentes a Sistemas de Recomendação. O Capítulo 3, que apresenta conceitos de música digital, uma breve história da gravação sonora, além das aplicações de transmissão de música e seus sistemas de recomendação e o estado da arte na área. O Capítulo 4, que apresenta a proposta: um sistema de recomendação de músicas baseado em conteúdo. Para além de discutir algoritmos e de como foi realizada a implementação. O Capítulo 5 apresenta a avaliação do sistema, e o Capítulo 6 apresenta as conclusões, considerações finais e trabalhos futuros.

⁵<http://labrosa.ee.columbia.edu/millionsong/>

2

Sistemas de Recomendação

Antes de vim, vi que esqueceram a soul train, a travelling man, as slows jam

—MARCELLO GUGU (Gil Scott Heron)

Como já introduzido por este trabalho, a grande número de informação produzida e disponibilizada atualmente pode gerar uma sobrecarga para o usuário. Por causa deste problema, muitas tecnologias surgiram, no intuito de para apoiar a seleção, recuperação e filtragem da informação de interesse do usuário. Este capítulo fornece uma visão geral sobre Sistemas de Recomendação, introduz os principais conceitos, as tarefas, técnicas que a caracterizam além dos prós e contras das técnicas abordadas.

2.1 Histórico

Os primeiros sistemas de recomendação foram tradicionais sistemas de filtragem e recuperação da informação (“*Information Filtering*” e “*Information Retrieval*”), os quais não podiam recomendar mais do que certos resultados de acordo com a pesquisa. O primeiro sistema de recomendação foi um sistema experimental de filtragem de *e-mail* denominado *Tapestry* (Goldberg *et al.*, 1992), desenvolvido por pesquisadores da *Xerox Palo Alto Research Center*. A motivação para o desenvolvimento do *Tapestry* veio do aumento do número de *e-mails* que chegavam ao centro de pesquisa (Resnick and Varian, 1997). *Tapestry* foi elaborado para lidar com filtragem colaborativa - sendo os primeiros a usar o termo - e filtragem baseada em conteúdo. A parte colaborativa do sistema filtra e arquiva os *e-mails* de acordo com as reações das pessoas que já tinham lido. Dois anos depois do *Tapestry*, (Shardanand, 1994) se propôs o “*Ringo*”, um sistema de recomendação de músicas que utiliza a filtragem social para recomendar músicas aos usuários através de *e-mails*. O usuário enviava uma lista com 10 álbuns através e algum tempo depois recebia uma lista com músicas que poderiam ser de seu interesse.

Em sua forma mais simples, recomendações personalizadas são fornecidas como uma lista ordenada de itens candidatos. Para encontrar os itens que constituirão essa lista, sistemas de

recomendação tentam prever qual produto ou serviço mais agradará aos usuários, baseando-se nas suas preferências. Para isso, eles coletam os dados, que podem ser expressas explicitamente, por exemplo, nas avaliações de produtos ou serviços. Ou são inferidas interpretando as ações do usuário no sistema. O sistema pode considerar um acesso à página de um produto como um sinal implícito de preferência ou até comentário sobre um produto, por exemplo. (Ricci *et al.*, 2011).

As pessoas frequentemente dependem de recomendações de outras, seja através de amigos, críticas de livros, filmes e músicas, ou guias de restaurantes e viagens. Resnick and Varian (1997) mostram que os sistemas de recomendações apoiam e aumentam esse processo natural. Basicamente, o que temos aqui são técnicas e ferramentas de programas para fornecer sugestões de itens que sejam úteis para um usuário. Nesses sistemas, “item” é um termo genérico utilizado para denotar o que o sistema oferece a um usuário (Ricci *et al.*, 2011), neste trabalho um “item” é uma música.

Sistemas de recomendação são largamente utilizados no comércio eletrônico, no entretenimento, no consumo de conteúdo, na indústria de serviço, e já podem ser encontrados em muitas aplicações contemporâneas, que entregam ao usuário uma enorme coleção de itens, recomendando-os ao usuário final utilizando combinações de abordagens como: baseada em conteúdo, filtragem colaborativa e em conhecimento (Ricci *et al.*, 2011). Por exemplo, o serviço de transmissão de filmes *Netflix*¹ prediz uma classificação para cada filme exibido, ajudando o usuário a decidir qual filme assistir. A loja online *Amazon*² fornece, na página de um produto, informações de outros produtos comprados por usuários que compraram aquele produto (Shani and Gunawardana, 2009). O serviço de transmissão de música *Spotify*³ recomenda músicas de acordo com as preferências do usuário. No capítulo a seguir, serão abordados as aplicações comerciais e o estado da arte da recomendação de música.

Com o rápido crescimento da quantidade de informação disponível na *Web* e o surgimento de vários novos tipos de serviços que, frequentemente, sobrecarregam o usuário (compra de produtos, comparação de produtos, notícias, etc), o usuário pode ficar perdido e pode vir a tomar decisões precipitadas ou erradas. Assim, a disponibilidade de escolhas ao invés de ser um benefício acaba sendo um problema para este usuário (Ricci *et al.*, 2011). Os sistemas de recomendação oferecem novos itens não visitados anteriormente como produtos, filmes, livros, páginas *Web*, músicas, etc, baseados nas preferências do usuário mantidas em seu perfil. Os sistemas de recomendação lidam com esse problema de sobrecarga de informação filtrando itens que podem corresponder aos interesses personalizados individualmente.

Critérios como “personalizada” e “interessante e útil” que diferenciam o sistema de recomendação dos sistemas de recuperação de informação, filtragem de informação ou motores de busca. Em um motor de busca, o sistema deve retornar tudo o que for correspondente aos termos da pesquisa. Porém, motores de buscas começaram a utilizar técnicas como Realimentação de

¹<https://www.netflix.com>

²<http://www.amazon.com/>

³<http://www.spotify.com/>

Relevância (**RR**) (em inglês, *Relevance Feedback* - RF) para introduzir o usuário no processo da busca, permitindo que os mesmos indiquem o que é realmente relevante (**Burke, 2002**).

2.2 Conceitos

Os sistemas de recomendação utilizam basicamente três tipos de dados para suprir seu objetivo: itens, usuários e relações dos usuários com os itens (**Ricci et al., 2011**).

- **Itens:** São os objetos recomendados. Podem ser caracterizados por sua complexidade e por seu valor ou utilidade. O valor deste pode ser positivo se o item é útil para o usuário, ou negativo se o item é inapropriado e o usuário tomou uma decisão errada ao selecioná-lo. Itens podem ser representados utilizando várias abordagens de representação e informação. Por exemplo, o filme é o item a ser recomendado no *Netflix* e a faixa no *Spotify*.
- **Usuários:** Podem ter diversos objetivos e características. Para personalizar as recomendações, os sistemas de recomendação exploram um conjunto de informações sobre o usuário. Essas informações podem ser estruturadas de várias maneiras, e a seleção de quais informações modelar depende da técnica de recomendação. Usuários também podem ser descritos pelos seus padrões de comportamento, como dados de navegação em um *site*.
- **Transações:** Transações são registros de interação entre o usuário e o sistema de recomendação. São como dados de log que armazenam informações importantes geradas durante a interação humano-computador, os quais serão usados pelos algoritmos de recomendação do sistema para encontrar recomendações úteis aos usuários.

2.3 Tarefas realizadas por um Sistema de Recomendação

Ricci et al. (2011) diferencia as tarefas dos Sistemas de Recomendação dos serviços que fazem uso das recomendações. Assim como, apresenta uma lista dos objetivos de um sistema de recomendação baseado nos interesses do criador do sistema de recomendação.

- **Aumentar a satisfação do usuário:** Uma combinação de recomendações precisas e uma interface bem elaborada irão aumentar positivamente a avaliação subjetiva do usuário a respeito do sistema. Isto irá aumentar a frequência de utilização do sistema e a probabilidade de as recomendações serem aceitas.
- **Aumentar a fidelidade do usuário:** Quando um usuário é fiel a um *site*, o mesmo deve reconhecê-lo como um cliente antigo e tratá-lo como um visitante de grande valor. Esse é um recurso frequente em sistemas de recomendação, já que eles encontram recomendações após avaliar as informações adquiridas sobre o usuário

em interações anteriores. Assim, quanto mais o usuário interage com o *site*, mais o modelo será refinado obtendo melhores recomendações.

- **Aumentar o número de itens vendidos:** Esta é, provavelmente, a função mais importante para um sistema de recomendação comercial. Ele deve ser capaz de vender um conjunto de itens adicionais comparado a aqueles que venderiam sem qualquer tipo de recomendação. Isto é devido ao sistema recomendar itens que são similares aos que os usuários desejam.
- **Vender mais itens diferentes:** Outra função importante de um sistema de recomendação é possibilitar que o usuário encontre itens úteis que seriam difíceis de encontrar sem uma recomendação precisa.
- **Melhor entendimento do que o usuário quer:** A descrição das preferências do usuário, coletas explicitamente ou implicitamente pelo sistema, podem ser reutilizadas para vários objetivos.

[Herlocker *et al.* \(2004\)](#) define onze tarefas de usuário que um sistema de recomendação pode auxiliar a implementar para apoiar o usuário em suas escolhas.

- **Anotação em Contexto:** Dada uma lista de itens em um contexto, evidenciar alguns deles de acordo com as preferências do usuário.
- **Encontrar Alguns Bons Itens:** Recomendar ao usuário uma lista ranqueada dos itens recomendados, com uma previsão do quanto os usuários gostariam dos itens. Esta é a principal tarefa dos sistemas de recomendação comerciais.
- **Encontrar Todos os Bons Itens:** Recomendar ao usuário uma lista de todos os itens que podem ser “úteis”. Em alguns casos encontrar apenas alguns bons itens pode não ser o suficiente. Na prática, o sistema tem que garantir ter uma taxa de “falso negativo” suficientemente baixa.
- **Recomendar uma Sequência:** Aqui se tem o desafio de recomendar uma sequência de itens relacionados.
- **Recomenda um Grupo:** Recomendar um grupo de itens relacionados que possam interessar o usuário.
- **Apenas Navegar:** O sistema deve auxiliar os usuários que apenas navegam sem um propósito definido, navegando entre os itens que possam ser interessantes naquela sessão específica.
- **Encontrar Sistema de Recomendação Confiável:** Os usuários podem não confiar no sistema de recomendação, assim o sistema deve possibilitar que o usuário teste algumas funções.

- **Melhorar o Perfil:** O sistema deve receber informações do usuário sobre o que ele gosta e não gosta. As contribuições do usuário irá aperfeiçoar a qualidade das recomendações.
- **Expressar-se:** Alguns usuários não se importam com as recomendações, apenas estão interessados em contribuir com as qualificações. Assim, o sistema deve fornecer uma seção para comentários.
- **Ajudar os Outros:** Alguns usuários gostam de contribuir com informação porque eles acreditam que a comunidade se beneficia de sua avaliação.
- **Influenciar os Outros:** Em Sistemas de Recomendação que utilizam a Internet, existem usuários maliciosos, que tentarão influenciar outros a escolher um determinado item, ou ainda utilizarão o sistema apenas para promover ou penalizar um item.

2.4 Técnicas de Recomendação

Para encontrar as recomendações que serão colocadas em forma de lista ranqueada, são utilizadas algumas técnicas, feitas a partir da predição sobre as informações dos itens e usuários.

Várias foram as técnicas propostas como base para um sistema de recomendação, como as técnicas colaborativa, baseada em conteúdo, baseada em conhecimento, demográfica, social e grupal. As técnicas de recomendação podem ser diferenciadas com base em suas fontes de dados: de onde vêm os dados necessários para encontrar as recomendações? Em alguns sistemas, esses dados são as preferências de outros usuários (Burke, 2007). Em outros sistemas, esses dados são as descrições que compõem o “item”.

A utilidade de um item para um usuário pode ser influenciada pelo conhecimento que este tem do domínio - por exemplo, usuários iniciantes versus experientes de uma câmera digital - ou pode depender do momento (local ou hora) em que a recomendação foi feita. Ou Suponhamos que o interesse maior seja por itens mais perto de sua localização - um restaurante ou local turístico por exemplo. Assim, o sistema de recomendação deve se adaptar para que esses detalhes específicos possam ser usados no momento de encontrar as recomendações (Ricci *et al.*, 2011).

2.4.1 Filtragem Colaborativa

Na filtragem colaborativa, como é possível verificar na Figura 2.1, o sistema recomenda ao usuário ativo itens que outros perfis com gostos semelhantes consumiram no passado. A semelhança de gostos de dois usuários é calculada baseada na similaridade do histórico de classificações de ambos. A filtragem colaborativa é considerada a técnica em sistemas de recomendação mais popular e mais largamente implementada (Ricci *et al.*, 2011).

A ideia chave é que a classificação de um usuário u para um novo item i é provável de ser similar com a de outro usuário v , se u e v têm classificado outros itens de maneira similar. De

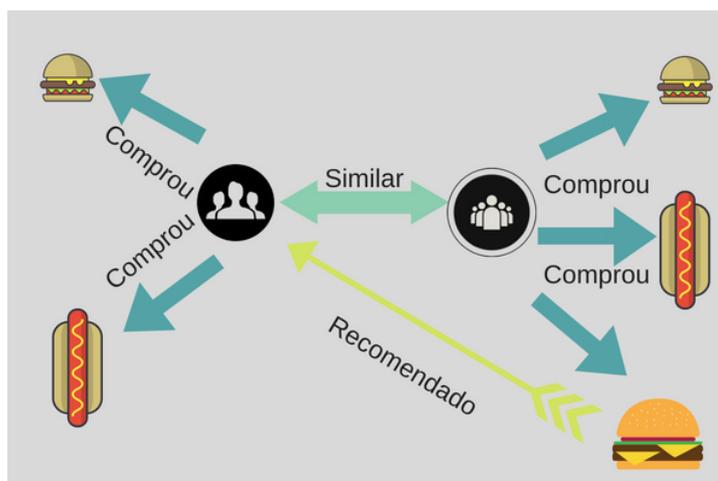


Figura 2.1: Exemplificação do método de Filtragem Colaborativa.

mesmo modo, é provável que u classifique dois itens i e j de forma semelhante, se outros perfis têm feito classificações aproximadas para estes dois itens (Ricci *et al.*, 2011).

Em Burke (2002) e Burke (2007), tem-se que os métodos de filtragem colaborativa podem ser divididos nas duas classes gerais: baseados em *vizinhança* e *modelo*. Na filtragem colaborativa baseada em vizinhança, as classificações usuário-item armazenadas no sistema são utilizadas diretamente para prever classificações de novos itens. Isto pode ser feito de duas maneiras conhecidas; se baseada no usuário ou no item (Ricci *et al.*, 2011). Já a que se baseia em modelo usa os dados do histórico do usuário para prever os itens que lhe serão relevantes.

Sistemas baseados no usuário avaliam o interesse de um usuário u por um item i utilizando as classificações, para este item, de outros usuários, chamados *vizinhos*, que têm padrões de classificação similar. Os vizinhos do usuário u são tipicamente os usuários v , cujas classificações para os itens categorizados por u e v , isto é I_{uv} , são mais correlacionados com aqueles de u . A abordagem baseada no item, por outro lado, prediz a classificação de um usuário u para um item i baseado nas classificações de u para itens similares a i . Em tais abordagens, dois itens são similares se muitos usuários do sistema tem classificado estes itens de maneira similar (Ricci *et al.*, 2011).

Diferente dos sistemas baseados em vizinhança, que usam as classificações armazenadas diretamente na predição, abordagens baseadas em modelo usam essas classificações para aprender um modelo preditivo. A ideia geral é modelar as interações usuário-item com fatores representando características latentes destes no sistema, como a classe de preferência do usuário e a classe de categoria dos itens. Este modelo é então treinado utilizando os dados disponíveis, e depois utilizado para prever classificações de usuários para novos itens (Ricci *et al.*, 2011).

2.4.2 Filtragem Baseada em Conteúdo

De acordo com Adomavicius and Tuzhilin (2005), o método de baseado em conteúdo tem sua origem nas pesquisas de recuperação da informação. Sistemas de recomendação baseado

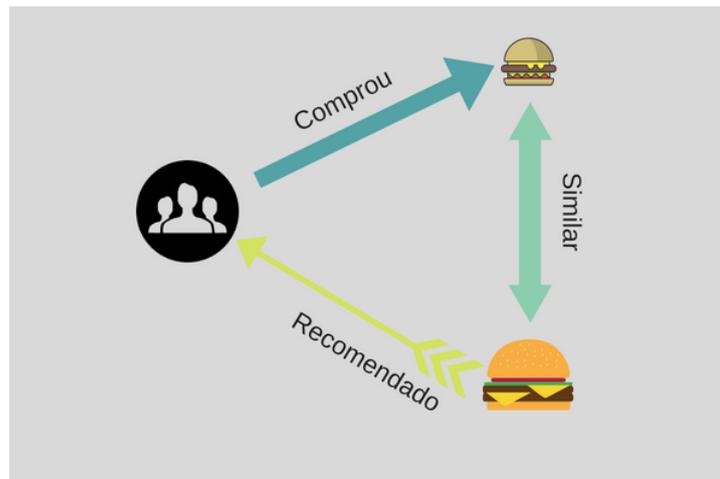


Figura 2.2: Exemplificação do método de Filtragem Baseada em Conteúdo.

em conteúdo indicam itens similares a aqueles que um usuário preferiu no passado, ao passo que sistemas que implementaram a técnica de recomendação colaborativa identificam usuários que possuem preferências similares a outros e sugere itens que eles têm preferido (Lops *et al.*, 2011). Na Figura 2.2 é possível visualizar que os usuários que compraram um item serão recomendados com outro item similar, por exemplo.

Sistemas que implementam uma abordagem de recomendação baseada em conteúdo analisam um conjunto de descrições dos itens previamente classificados pelo usuário u , para assim construir um perfil ou modelo de interesses do usuário baseado nas características dos itens classificados (Mladenic, 1999). O modelo é uma representação estruturada das preferências do usuário, adaptado para encontrar e recomendar novos itens que possam ser de seu interesse. O processo de recomendação consiste em comparar os atributos do modelo do usuário com os atributos de um item. O resultado é um julgamento de relevância que representa os níveis de interesse do usuário naquele item. Se um modelo reflete com precisão as preferências do usuário, isso é uma grande vantagem para a eficácia no processo de recomendação (Lops *et al.*, 2011).

2.4.3 Arquitetura de Sistemas de Recomendação Baseados em Conteúdo

Segundo Lops *et al.* (2011), em sistemas baseados em conteúdo é necessário utilizar técnicas apropriadas para representar ou anotar os itens e produzir o modelo do usuário, e algumas estratégias para compará-lo com a representação do item. O processo de recomendação é realizado em três passos, cada qual é manipulado por um componente separado:

- **Analisador de Conteúdo:** Quando a informação não tem estrutura, como um texto, algum tipo de pré-processamento é preciso para extrair informação relevante e estruturada. A principal responsabilidade do componente é representar a descrição dos itens (por exemplo, áudio, documentos, páginas *Web*, notícias, descrição de produtos, etc.) vindos de fontes de informação de forma adequada. Esta informação extraída é a entradas dos próximos passos.

- Aprendiz de Perfil:** Este módulo coleta dados representativos das preferências dos usuários e tenta generalizá-los, para construir o modelo do usuário. Normalmente, a estratégia de generalização é realizada através de técnicas de aprendizado de máquina, que são capazes de inferir um padrão de interesses, partindo de itens preferidos ou não preferidos no passado. Por exemplo, recomendadores podem implementar o método de retorno do usuário, onde a técnica de aprendizado combina os retornos positivos e negativos do usuário para assim poder modelar seu perfil.
- Componente de Filtragem:** Este módulo explora o modelo do usuário para sugerir itens relevantes através da combinação da sua representação contra os itens a serem recomendados. O resultado é um binário ou contínuo julgamento de relevância - computado utilizando alguma métrica de similaridade (Herlocker *et al.*, 2004) -, neste último caso resultando em uma lista ranqueada de itens potencialmente interessantes.

Em seu trabalho Lops *et al.* (2011) exibe uma arquitetura de alto nível para um sistema de recomendação baseado em conteúdo, assim como é possível visualizar na Figura 2.3.

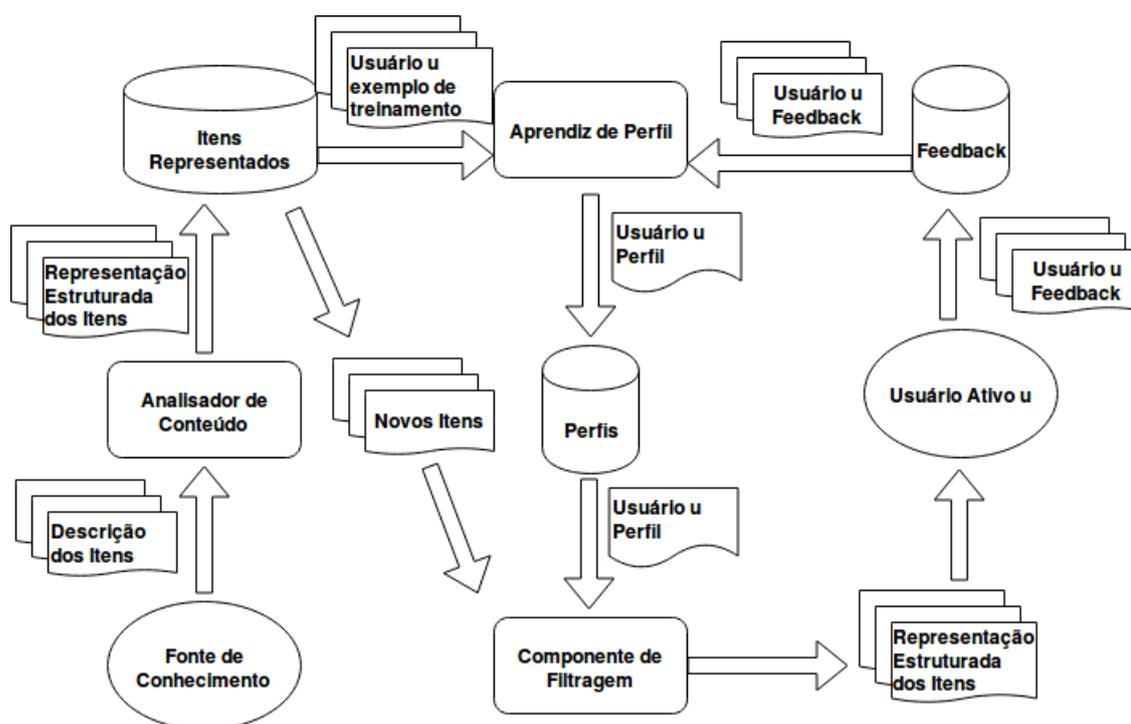


Figura 2.3: Arquitetura de alto nível de um Sistema de Recomendação Baseado em Conteúdo (Lops *et al.*, 2011).

Celma (2014) apresenta duas formas de analisar os conteúdos das músicas: automática ou manualmente. Geralmente, no processo automático de extração das características, o áudio é semanticamente anotado para que então a similaridade entre as músicas seja computada. Depois disso, músicas são recomendadas se “soarem” de forma semelhante às músicas no modelo do usuário. Em relação à extração manual, apesar de ser considerada muito demorada, devido à

entrada de dados ser dependente de humanos, pode ser ainda mais precisa do que o processo automático.

2.4.4 Prós e Contras das Técnicas de Recomendação

Todas as técnicas de recomendação tem seus pontos fortes e fracos. [Isinkaye et al. \(2015\)](#) realizam um levantamento entre as mais usadas descrevendo prós e contras.

- **Filtragem Baseada em Conteúdo:** Essas técnicas têm a capacidade de recomendar novos itens, mesmo se não houver muitas classificações fornecidas pelos usuários. Portanto, mesmo que o modelo do usuário não contenha um grande número de preferências, a precisão da recomendação não é afetada. Além disso, se as preferências do usuário mudarem, ele terá a capacidade de ajustar suas recomendações em um curto espaço de tempo. Elas podem gerenciar situações em que usuários diferentes não compartilham os mesmos itens, mas apenas itens idênticos. Os usuários podem obter recomendações sem compartilhar seu modelo e isso garante a privacidade como no estudo de [Shyong et al. \(2006\)](#) que aborda sobre exposição dos dados pessoais, manipulação do enviesamento do usuário e sabotagem para reduzir a eficiência do acerto em sistemas de recomendação. A técnicas baseadas em conteúdo também pode fornecer explicações sobre como as recomendações são encontradas para os usuários. No entanto, as técnicas sofrem de vários problemas, como discutido na revisão da literatura ([Adomavicius and Tuzhilin, 2005](#)). As técnicas de filtragem baseadas em conteúdo dependem dos metadados ou anotações dos itens. Ou seja, eles exigem uma rica descrição de itens e um modelo de usuário bem organizado antes que a recomendação possa ser realizada. Isso é chamado de análise de conteúdo limitada. Assim, a eficácia desta técnica depende da disponibilidade de dados descritivos. A superespecialização de conteúdo é outro problema grave, os usuários estão restritos a receber recomendações semelhantes aos itens já definidos em seus modelos. Outro problema é quando o usuário não tem preferências no sistema precisando assim avaliar um número suficiente de itens antes que o sistema possa realmente entender as preferências do usuário e apresentar ao usuário recomendações “confiáveis”.
- **Filtragem Colaborativa:** Estas técnicas podem ser executadas em domínios onde não há muito conteúdo associado a itens ou onde o conteúdo é difícil de análise (como opiniões, ideias e sentimentos). Além disso, tem a capacidade de fornecer recomendações inesperadas, o que significa a possibilidade recomendar itens desconhecidos que sejam relevantes para o usuário, mesmo sem algum conteúdo semelhante estar presente no modelo do usuário ([Schafer et al., 2007](#)). Apesar do sucesso das técnicas de filtragem colaborativa, seu uso difundido revelou alguns problemas potenciais. Problema de dispersão ou esparsidade de dados é o problema que ocorre como resultado da falta de informações suficientes, ou seja, quando uma pequena parcela

dos itens disponíveis no sistema são classificados pelos usuários (Burke, 2002). Isso sempre leva a uma matriz esparsa de itens no modelo do usuário, incapacidade de localizar vizinhos bem-sucedidos e, finalmente, a geração de recomendações não “confiáveis”. Outro problema é do usuário novo no sistema, que é o mesmo problema dos sistemas baseados em conteúdo. Para fazer recomendações precisas, o sistema deve primeiro criar um modelo do usuário a partir das classificações fornecidas pelo mesmo. Várias técnicas foram propostas para resolver este problema. A maioria deles usam as abordagens de recomendação híbrida, que combina técnicas colaborativas e baseadas em conteúdo.

2.5 Sumário

Neste capítulo, vimos uma visão geral sobre os Sistemas de Recomendação. Na Seção 2.1 começamos analisando o histórico sobre Sistemas de Recomendação. Em seguida na Seção 2.2 discutimos alguns conceitos sobre os dados utilizados em um sistema. Então, na Seção 2.3 foi apresentado uma lista de tarefas desempenhadas por sistemas de recomendação. Por fim, na Seção 2.4 discutimos as principais técnicas de recomendação assim como seus prós e contras.

3

Música Digital

A música é a afinação da interioridade.

—GILBERTO GIL

O fonógrafo (Wikipedia, 2018; Globo, 2012) construído pelo francês Léon Scott em 1857 foi o primeiro aparelho a gravar ondas sonoras sem a intenção de ser reproduzida, usada apenas para análise visual das ondas no papel. Durante os seus testes em 1860 a primeira música gravada foi "*Au Clair de la Lune*", de Pierrot Répondit. Em 1877, Thomas Edison construiu o Gramofone (em inglês, *Phonograph*) aparelho capaz de gravar e reproduzir ondas sonoras, o mesmo Thomas gravou a música "*Mary had a little lamb*". Já no Brasil, a primeira música gravada foi "Isto é Bom" de Xisto Bahia em 1902 (Mix, 2013). Com o passar dos anos a indústria da música evoluiu, sempre buscando uma melhor qualidade nas reproduções e assim, em 1963, a Fita Cassete foi desenvolvida pela *Philips*¹. Em 1982, a era da música digital iniciou-se com o CD, desenvolvido em parceria pela *Philips* e *Sony*². Em 1993, o *Internet Underground Music Archive* (IUMA) foi lançado como o primeiro acervo de músicas on-line gratuito para download em MP3. Mas foi em 2002 que a primeira companhia de transmissão de música utilizando sistema de recomendação apareceu. O *Last.fm* foi lançado como uma plataforma que usa um recomendador de música para criar um perfil detalhado do gosto musical de cada usuário após registrar os detalhes das faixas que eles ouvem. Sendo seguido em 2005 pelo *Pandora*, 2007 pelo *Soundcloud* e em 2008 pelo *Spotify*.

3.1 Introdução

Os formatos digitais de áudio proporcionaram a distribuição maciça de músicas pela Internet. Em seu início na Web, as faixas musicais eram vendidas em comércios eletrônicos (*E-Commerce*), facilitando o acesso e a descoberta de novidades pelos usuários. Com a melhoria na qualidade da Internet, tanto em velocidade quanto em confiabilidade da entrega, os serviços

¹<https://www.philips.com.br/>

²<https://www.sony.com.br/>

de transmissão de música (*Music Streaming*) começaram a aparecer, disponibilizando milhares de músicas em sistemas especializados na descoberta de novos itens pelos usuários. De acordo com [IFPI \(2018\)](#), em 2017 o *Download* de músicas caiu 20,5% assim como a venda do álbum físico que caiu 5,4%, em contrapartida o lucro global da indústria musical cresceu 8,1% sendo puxado pelo crescimento de 45,5% dos usuários pagos nos serviços de transmissão de música, sendo que neste mesmo ano 54% do lucro da indústria provém desses serviços.

3.2 Aplicações com Sistemas de Recomendação de Música

A popularização dos serviços de comércio eletrônico e transmissão de música, facilitou bastante o acesso a novas descobertas, mas também trouxe uma carga a mais para o usuário na procura do que pode lhe satisfazer. Com isso em vista, tais serviços aderiram aos sistemas de recomendação para focar o indivíduo nas canções que possam vir a ser do seu agrado, baseando suas decisões em diversas informações e conhecimentos que esses sistemas vão obtendo sobre os usuários.

3.2.1 E-Commerce

Os comércios eletrônicos (*E-Commerce*) em sua maioria seguem o modelo de negócio da venda de álbuns em [CD](#) ou uma cópia digital do álbum, normalmente seus sistemas trabalham com outros itens além do domínio musical. Assim recomendar esses sistemas utilizam informações de metadados dos itens, usuários e suas ações na Web. Sendo as recomendações geradas por raciocínios genéricos sobre as relações dos itens e com os usuários.

- **Amazon:** A *Amazon*³ um dos principais comércios eletrônicos do mundo após a implementação do sistema de recomendação, suas compras cresceram 29% em apenas três meses ([Hint, 2016](#)). A recomendação de músicas ao usuário é realizada através de álbuns com maiores notas, exemplificando assim na [Figura 3.1](#) uma implementação do método de filtragem colaborativa.
- **Submarino:** O *Submarino*⁴, comércio eletrônico brasileiro, realiza a venda de diversos itens, entre eles os álbuns musicais em [CD](#). Na [Figura 3.2](#) é possível visualizar que o método de filtragem colaborativa baseado nos itens foi adotado para gerar as recomendações dos mais bem avaliados, para um usuário ao qual o sistema não tem conhecimento prévio.

³<https://www.amazon.com.br/>

⁴<https://www.submarino.com.br/>

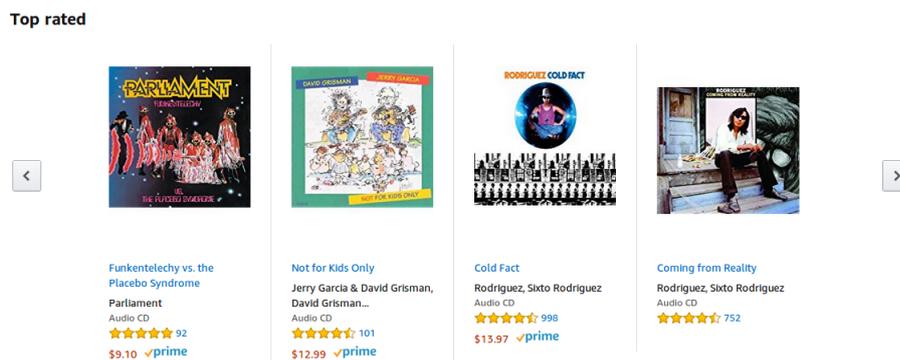


Figura 3.1: Recomendações de álbuns no comércio eletrônico da Amazon.

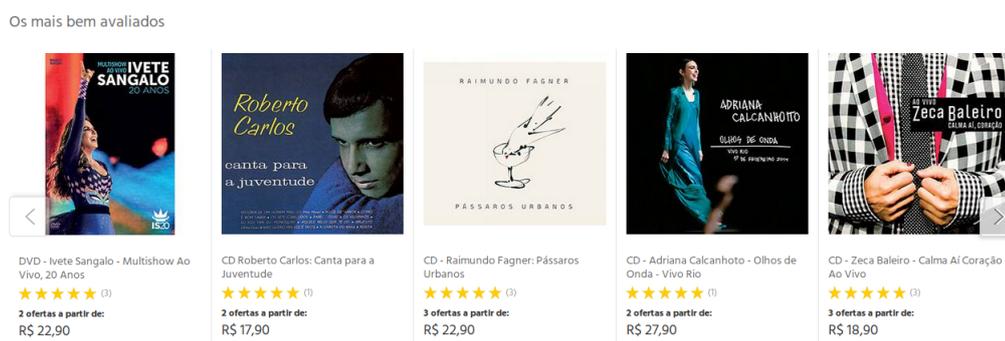


Figura 3.2: Recomendações de álbuns musicais no comércio eletrônico Submarino.

3.2.2 Music Streaming

Atualmente os serviços de transmissão de música (*Music Streaming*) ocupam o primeiro lugar em difusão musical pela Internet, assim como o principal arrecadador financeiro da indústria musical. Tais serviços contam também com um sistema de recomendação, utilizando as mais variadas técnicas possíveis.

- **Spotify:** O *Spotify*⁵, o serviço de transmissão de música mais usado, implementa diversas técnicas de recomendação, na Figura 3.3 uma lista com novas músicas para o usuário, baseando-se nos gostos do mesmo, a lista é refeita semanalmente sempre com 30 novas faixas, exemplificando uma implementação de filtragem baseada em conteúdo, já a Figura 3.4 o *Spotify* recomenda as mais ouvidas no Brasil, país onde o usuário se encontra, exemplificando uma implementação Demográfica.
- **Last.fm:** O *Last.fm*⁶, é um dos serviços de transmissão de música mais antigos. Criado em 2002, implementa diversos métodos de recomendação mas ganhou fama utilizando método de filtragem colaborativa. Na Figura 3.5, é possível verificar a implementação do método de Híbrida, o uso das *Tags* para agrupar junto com as mais ouvidas pelos usuários. Já a Figura 3.6 mostra quatro formas de utilizar o método de filtragem colaborativa.

⁵<https://www.spotify.com/>

⁶<https://www.last.fm/pt/>

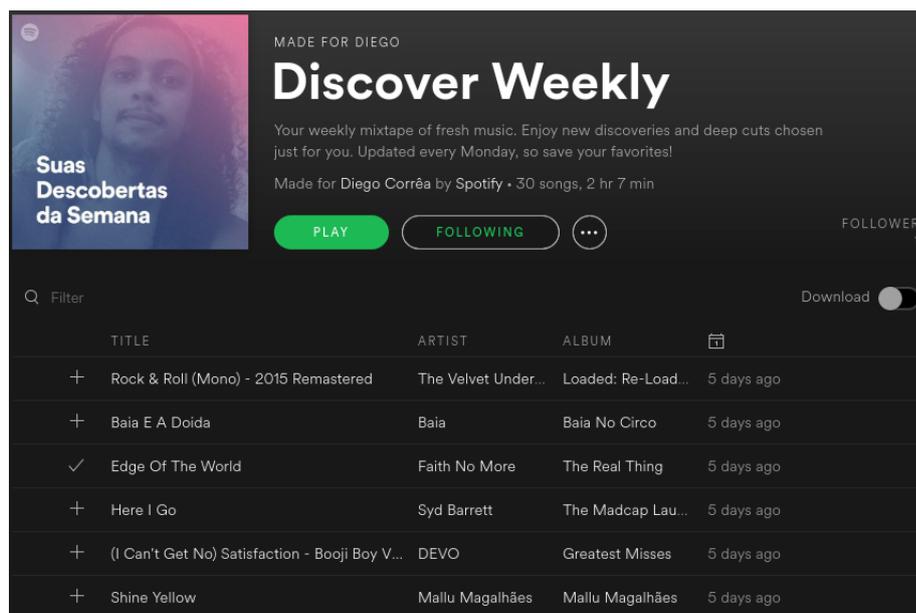


Figura 3.3: Lista de recomendações semanal do *Spotify* com 30 novas músicas para serem descobertas na semana.

- Youtube:** Em seu serviço, o *site* promove não somente *Streaming* de áudio, mas também de vídeo. Artistas e gravadoras comumente lançam videoclipes para promover seus lançamentos. Assim o serviço veio a ser conhecido devido ao seu grande acervo de músicas e *Clips*. O *Youtube* conta com um sistema de recomendação genérico onde os itens a serem recomendados são os vídeos. A Figura 3.7 mostra as recomendações personalizadas para o usuário, a partir dos vídeos previamente visualizados pelo menos, mostrando assim uma abordagem baseada em conteúdo.

3.3 Sistema de Recomendação de Música

De acordo com [Celma \(2014\)](#) a pesquisa sobre recomendação de música não começou a sério até 2000, com uma breve exceção ([Shardanand, 1994](#)). A recomendação de música é uma área de pesquisa interdisciplinar, que envolve desde de pesquisa e filtragem a musicologia, mineração de dados, aprendizado de máquina, personalização, redes sociais, processamento de texto, redes complexas, interação do usuário, visualização de informações e processamento de sinais ([Celma and Lamere, 2011](#)). Assim, permite-se aplicar variadas técnicas de recomendação, bem como utilizar os mais variados tipos de dados.

3.3.1 Filtragem Colaborativa

Sistemas de recomendação de música tradicionais usam filtragem colaborativa. O primeiro estudo foi o Ringo [Shardanand \(1994\)](#), um sistema de recomendação de música via *e-mail*, onde os usuários classificavam com notas os artistas e assim recebiam sugestões. [Lemire](#)



Figura 3.4: Lista de recomendações de músicas do *Spotify* baseada no país do usuário.



Figura 3.5: Recomendações de estilos musicais do *Last.fm* baseada em *Tags*.



Figura 3.6: Utilização do método de filtragem colaborativa baseado em item pelo *Last.fm*.

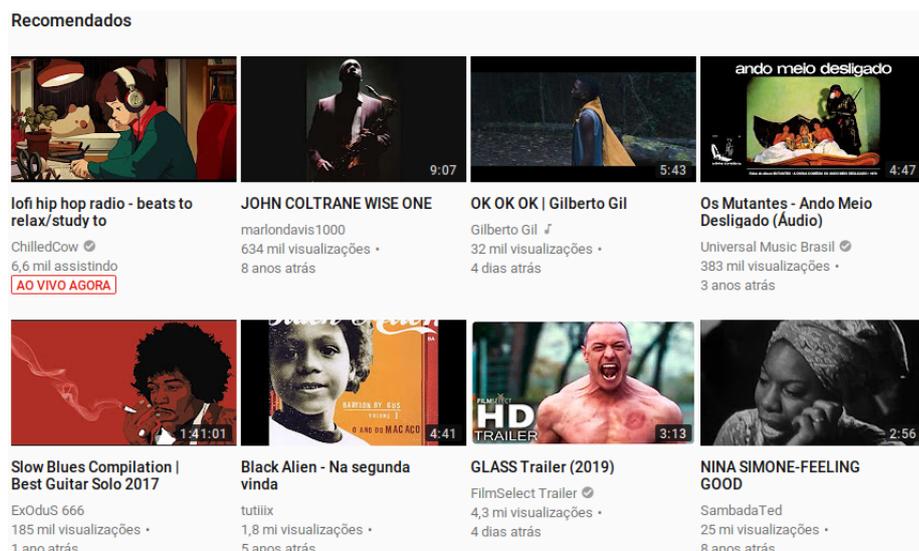


Figura 3.7: Lista de recomendações do *Youtube* contendo músicas e trailers de filme.

and Maclachlan (2005) exploraram a filtragem colaborativa baseada na classificação do usuário como um processo de prever como o mesmo classificaria uma determinada música a partir de outras avaliações dos usuários, para isso utilizaram a distância dada pelo Coeficiente de Correlação de Pearson.

3.3.2 Filtragem Baseado em Conteúdo

Logan (2004) explora o problema da recomendação de música baseada exclusivamente na frequência do áudio, onde o áudio é convertido em Mel-Frequency Cepstral Coefficients (MFCCs), que permite assim capturar informações espectrais suavizadas que correspondem aproximadamente a instrumentação e timbres. Em seu trabalho, ele também propõe soluções que podem ser usadas com qualquer medida de similaridade baseada em acústica: a mediana, a média e a distância mínima da similaridade entre uma música e um conjunto de músicas. Hijikata *et al.* (2006) propõe um sistema de filtragem, baseado em conteúdo, usando uma árvore de decisão. Utilizando diversos dados contidos na extensão de áudio Musical Instrument Digital Interface (MIDI), como: tempo, tonalidade, ritmo, tom, metro, *pitch*, a diferença de medula e a diferença de duração. Knees *et al.* (2006) apresentam uma técnica para combinar a semelhança musical do sinal de áudio com semelhança musical do artista, baseado na web para acelerar a tarefa de geração automática de listas de reprodução. Explorando a similaridade determinada automaticamente por meio de recuperação da Web para evitar o cálculo de similaridade entre as faixas de diferentes artistas e/ou não relacionados.

3.3.3 Híbrida

Desde 2005, a comunidade de pesquisa ampliou sua atenção para incluir outros aspectos, como abordagem híbrida. Yoshii *et al.* (2006) apresentaram um método de recomendação de

música híbrido que resolve problemas de dois métodos convencionais proeminentes: filtragem colaborativa e recomendação baseada em conteúdo. O método usado integra dados de classificação e conteúdo, usando uma rede Bayesiana chamada modelo de aspecto. [Yoshii and Goto \(2009\)](#) apresentaram uma indexação semântica latente probabilística (pLSI), que lida com dados de classificação fornecidos pelo usuários e outros extraídos de sinais de áudio. [Tiemann and Pauws \(2007\)](#) investigaram os métodos de aprendizagem de conjunto (*Ensemble Learning*), combinando uma abordagem social e algoritmos de recomendação baseado em conteúdo. [Magno \(2008\)](#) implementou uma abordagem baseada na similaridade de sinais de áudio que foram reforçadas com base em suas correlações com os julgamentos humanos. Além de utilizar aplicações de *Streaming*, como: Allmusic, que usa rótulos de gêneros, Pandora, que usa análise musicológica, e Last.fm que usa colaborativa filtragem. [Hornung et al. \(2013\)](#) apresentaram uma abordagem de recomendação híbrida ponderada que combina três técnicas diferentes de recomendação. Utilizando-se de uma heurística de serendipidade que apimenta a lista de recomendações e um recurso de explicação que ajuda o usuário a entender a lógica por trás de cada uma das faixas recomendadas a ele.

3.3.4 Sensível ao Contexto

[Shin et al. \(2009\)](#) abordaram um sistema que explora as informações do contexto dos usuários para uma recomendação mais precisa. Como o contexto bruto dos usuários não pode ser aplicado diretamente aos sistemas de recomendação, ele foi abstraído em um nível de conceito, analisando a semântica das informações de contexto. [Pichl et al. \(2017\)](#) estudam como os agrupamentos contextuais podem ser aproveitados para recomendações de música, realizando a extração do contexto a partir do título das listas de reprodução.

3.3.5 Marcação Social

Marcações sociais (*Social Tags*) são palavras-chave geradas por usuários que são associadas a algum recurso ou acontecimento na Web. No caso da música, as *tags* sociais se tornaram um componente importante dos sistemas de recomendação, permitindo que os usuários criem listas de reprodução, como dormir ou praia, aplicados a determinadas músicas. [Eck et al. \(2008\)](#) propuseram um método para prever as *tags* sociais diretamente de arquivos MP3. Usando um conjunto de classificadores e *tags* sociais coletadas da Web associadas a música. Já [Symeonidis et al. \(2008\)](#) propõem uma abordagem para o problema fazendo-se uso de metadados como: usuários, *tags* e itens. Recomendando faixas de acordo com a percepção dos usuários sobre a mesma, realizando análise semântica latente e redução de dimensionalidade, utilizando a técnica de Higher Order Singular Value Decomposition (HOSVD).

3.3.6 Redes de Músicas

[Buldú et al. \(2007\)](#) desenvolveram um estudo de uma rede social construída sob a influência de gostos musicais. A rede é obtida da seguinte forma: quando duas obras coocorrem em uma lista de reprodução, um link entre elas é criado, levando a uma rede complexa onde elas são os nós. Nessa representação, as canções da mesma lista de reprodução podem pertencer a gêneros musicais diferentes, mas são propensas a estarem ligadas por um certo gosto musical (por exemplo, se as composições *A* e *B* coocorrem em várias listas de reprodução, um usuário que gosta de *A* provavelmente também *B*). [Seyerlehner et al. \(2009\)](#) mostraram que as músicas recomendadas são transformadas em um grafo de recomendação. Também analisaram as sugestões de um sistema real para descobrir se os usuários podem realmente explorar o banco de dados das peças subjacentes seguindo essas recomendações. [McFee and Lanckriet \(2009\)](#) apresentaram um sistema de recomendação de artistas que integra várias fontes de dados. O sistema utiliza atributos sociais, semânticos e de áudio, para construir um grafo de similaridade em nível de artista.

3.3.7 Visualização

[Goto and Goto, 2005](#)) desenvolveram uma interface de reprodução musical chamada Musicream que permite ao usuário um alto grau de liberdade e que encontre inesperadamente várias músicas semelhantes àquelas a que gosta. [Pampalk and Goto \(2006\)](#) apresentaram o MusicRainbow, que é uma interface para descobrir artistas onde as cores representam diferentes tipos de música. Os artistas semelhantes são mapeados próximos uns dos outros em um arco-íris circular. Também apresentam uma combinação de informações baseadas em atributos de áudio com dados da Web, rotulando o arco-íris e resumindo os artistas com palavras extraídas de páginas da Web.

3.3.8 Gerador de Lista de Reprodução

[Flexer et al. \(2008\)](#) apresentam um sistema gerador de listas de reprodução com base em uma música de início e fim, criando uma transição suave que permite aos usuários descobrirem novas músicas em uma coleção. A abordagem utilizada é baseada na similaridade dos atributos de áudio MFCCs e não requer nenhum tipo de metadado manualmente extraído. [Fields et al. \(2011\)](#) apresenta em seu trabalho um sistema que utiliza metadados das redes sociais e os atributos do áudio para criar uma lista de reprodução personalizada. [Pampalk et al. \(2005\)](#) apresentam um sistema que adapta automaticamente as listas de reprodução, a partir de uma música inicial e *feedback* imediato do usuário. Os atributos de áudio foram utilizados para encontrar a similaridade entre as músicas. O usuário dá o *feedback* pressionando o botão “pular” caso não goste da música atual. Músicas semelhantes as ignoradas são removidas, enquanto as que se assemelhem as não ignoradas são adicionadas à lista de reprodução.

3.3.9 Recomendação em Grupo

[Cho et al. \(2007\)](#) estudaram um sistema de recomendação de música baseado em uma rede de sensores em locais públicos, onde há uma grande movimento de pessoas, como: hotéis, parques, metrô, supermercado, etc., que são instalados em locais onde as pessoas estão indo e vindo com frequência. Para escolher uma música que melhor corresponda ao contexto atual, diversas variáveis, como: densidade de pessoas, estação do ano, clima e horário, são usadas como parâmetros. Uma comunicação dos sensores com o servidor permite saber o número de pessoa no local que ajudará na recomendação.

3.4 APIs e Datasets

Os sistemas de recomendação são diretamente alimentados com dados, sendo eles o carro chefe para a existência desses sistemas. Dentre as diversas formas de obter esses dados, a seguir iremos abordar duas: Conjuntos de dados (*Datasets*) e as Application Programming Interfaces (*APIs*).

3.4.1 Dataset

Conjuntos de dados são importantes tanto para pesquisadores quanto para aplicações comerciais que desejam testar seu sistema proposto. Assim diversos *Datasets* musicais foram disponibilizados gratuitamente na Internet. Alguns motivos de utilizar *Datasets* são:

- Para encorajar pesquisas em algoritmos de escala comercial.
- Um grande conjunto de dados pode revelar problemas de escala com o algoritmo usado que não foi identificado no período de teste do sistema.
- Prover referência de avaliação para o sistema proposto e comparação de resultados.
- Ajudar novos pesquisadores da área.

3.4.1.1 Million Song Dataset

O *Million Song Dataset* ([Mahieux et al., 2011](#); [McFee et al., 2012](#)) é uma coleção grátis de informações de áudio e metadados para um milhão de faixas musicais. O *Dataset* contém: 1.000.000 de músicas e arquivos, 44.745 artistas, 2.321 *tags*. Trazendo de cada canção um total de 55 campos de dados, como: título, ano, artista, letra, andamento, timbre, duração e etc. Conjuntos menores de dados com menos campos também são disponibilizados.

3.4.1.2 UCI

O *UCI*⁷ disponibiliza diversos conjuntos de dados para uma grande quantidade de domínios. Em particular no caso a música dois conjuntos de dados estão disponíveis: *FMA: A Dataset For Music Analysis Data Set* e *Geographical Original of Music Data Set*. O primeiro conta com 106.574 faixas de áudio (codificada como mp3). Com uma média 10 milhões de amostras por trilha. Nove recursos de áudio (consistindo em 518 atributos) para cada uma das 106.574 faixas. O segundo possui de um conjunto de dados com 1059 faixas, cobrindo 33 países, sendo composto de dois arquivos. O primeiro com 68 colunas que são recursos de áudio da faixa, e as duas últimas colunas são a origem da música, representada por latitude e longitude. O Segundo arquivo com 116 colunas de recursos de áudio da faixa, e as duas últimas colunas são a origem da música.

3.4.1.3 MusicBrainz

O *MusicBrainz*⁸ é um conjunto de dados voltado para a música com conteúdo aberto, com informações sobre artistas, seus trabalhos gravados e as relações entre eles. Os trabalhos contidos no *MusicBrainz* possuem, no mínimo, o título do álbum, os títulos das faixas e a duração de cada faixa. Atualmente o conjunto de dados possui aproximadamente 1,1 milhão de artistas, 1,6 milhão de lançamentos e 16 milhões de gravações.

3.4.2 APIs

Interface de Programação de Aplicações, cujo acrônimo *APIs* provém do Inglês Application Programming Interfaces, é um conjunto de rotinas e padrões, estabelecidos por um software, para a utilização das suas funcionalidades por aplicativos que não pretendem envolver-se em detalhes da implementação do software, mas apenas usar seus serviços. O uso das *APIs* nos sistemas de recomendação ajuda a aplicação que está iniciando a testar seus algoritmos e sua escalabilidade, utilizando dados de uma aplicação real, além de ser ponto de referência para medir os resultados obtidos de diferentes trabalhos. Atualmente, o padrão dos serviços de transmissão de música é o Representational State Transfer (*REST*) (Roy, 2000).

3.4.2.1 Last.fm

A *Last.fm*⁹ API permite que qualquer pessoa construa seus próprios programas usando dados Last.fm, se eles estão na web, na área de trabalho ou em dispositivos móveis. A API permite que você chame métodos que respondem no estilo *Extensible Markup Language - XML* usando o padrão de comunicação *REST*. A partir de endereços da Web é possível obter dados sobre: álbuns, artistas, usuários, geolocalização, *Tags* e atributos de áudio.

⁷<https://archive.ics.uci.edu/ml/datasets.html>

⁸<https://musicbrainz.org/>

⁹<https://www.last.fm/api/intro>

3.4.2.2 Spotify

O *Spotify* ¹⁰ tem como base os princípios [REST](#), os resultados da API do *Spotify Web* retornando metadados em *JavaScript Object Notation - JSON* sobre: artistas, álbuns, faixas musicais, listas de reprodução, canções que os usuários adicionaram nos favoritos.

3.5 Sumário

Neste capítulo introduziram-se os conceitos de música digital, uma breve história da gravação sonora, além das aplicações de transmissão de obras e seus sistemas de recomendação e o estado da arte da área. Os próximos capítulos estão organizados da seguinte maneira: O Capítulo 4 apresenta o Ouvido Musical um sistema de recomendação baseado em conteúdo, que sugere ao usuário músicas que podem vir a ser de sua preferência, além de uma discussão de como foi feita a implementação. O capítulo 5 apresenta a avaliação do sistema e no capítulo 6 conclusões e considerações finais.

¹⁰<https://developer.spotify.com/>

4

Ouvindo Musical: Um Sistema de Recomendação de Música

O Brasil tem ouvido musical. Que não é normal.

—CAETANO VELOSO (Love, Love, Love)

Para que um software seja desenvolvido de forma consistente, é preciso aliar boas práticas da engenharia de software com um robusto e eficiente processo de desenvolvimento. Diferentes tipos de sistemas necessitam de diferentes processos de desenvolvimento (Sommerville, 2010). Neste capítulo será abordada a solução proposta: *Ouvindo Musical* um sistema de recomendação de música baseado em conteúdo com recomendações personalizadas, baseadas nas preferências dos usuários, utilizando os metadados das músicas. Além de explicitar o processo de construção, traçando a arquitetura das camadas, apresentando tecnologias usadas no desenvolvimento, os algoritmos e as modelagens dos dados.

4.1 Algoritmos

Nesta seção serão apresentados os fundamentos teóricos dos algoritmos utilizados neste trabalho, assim como exemplos de uso.

4.1.1 Medidas de similaridade baseada em conteúdo

Neste trabalho, utilizaremos a extração manual de informações, ou seja, metadados textuais disponibilizados por algum conjunto de dados público. Assim, o sistema proposto poderá encontrar a similaridade entre os metadados, calculando o cosseno entre os dados usando o TF-IDF para compor o modelo vetorial.

4.1.2 TF-IDF

O *Term Frequency–Inverse Document Frequency* - (TF–IDF), que significa frequência do termo–inverso da frequência nos documentos, é uma medida estatística que tem o intuito de indicar a importância de uma palavra de um documento, em relação a uma coleção de documentos, ou em um corpus linguístico. O valor TF de uma palavra aumenta proporcionalmente à medida que aumenta o número de ocorrências dela em um documento. Esse valor, no entanto, é equilibrado pela frequência da palavra no corpus IDF . Isso auxilia a distinguir o fato da ocorrência de algumas palavras serem geralmente mais comuns que outras.

Na Equação 4.1, o TF é dado como n_t que é o número de vezes em que o termo t aparece no documento m dividido por n_m , que é o número total de termos no documento m . O IDF, na Equação 4.2, trabalha como um ajuste no TF, visto que alguns termos ocorrem mais que outros. Assim, na equação, n_M é dado como o número total de documentos pertencentes ao conjunto M , dividido por n_{Mt} , que é o número de documentos pertencentes ao conjunto M que contenha o termo t . O 1 adicionado no denominador previne a divisão por 0, o 1 adicionado no dividendo garante que o valor seja maior ou igual a 1, o terceiro 1 garante que o valor seja maior que 0. Assim, na Equação 4.3, temos a multiplicação do TF pelo IDF.

$$TF(t, m) = \frac{n_t}{n_m} \quad (4.1)$$

$$IDF(t, M) = \log\left(\frac{n_M + 1}{n_{Mt} + 1} + 1\right) \quad (4.2)$$

$$TFIDF(t, m, M) = \frac{n_t}{n_m} \times \log\left(\frac{n_M + 1}{n_{Mt} + 1} + 1\right) \quad (4.3)$$

Por exemplo, sendo o corpus M composto pelos documentos: “*Rock It*”, “*Easily*”, “*Evil Deeds*”, “*Rock a My Soul*”, “*Easily*”, “*Rock The Beat*”, “*Music My Rock*”, “*Robot Rock*”. Cada documento é processado pela biblioteca NLTK para a retirada de “palavras vazias” (*stop word*), vetorização, e identificação dos termos: “*beat*”: 0, “*deed*”: 1, “*easily*”: 2, “*evil*”: 3, “*music*”: 4, “*robot*”: 5, “*rock*”: 6, “*soul*”: 7. Palavras como: “*it*”, “*a*”, “*my*”, “*the*”, são retiradas durante o pré-processamento por serem palavras vazias. Sabendo-se agora quais são os termos do corpus M , torna-se viável realizar uma contagem dos termos em todos os documentos. A Tabela 4.1 mostra os resultados do TF para cada documento baseado na Equação 4.1, a Tabela 4.2 mostra o resultado do IDF para cada termo baseado na Equação 4.2 e a Tabela 4.3 contém o valor equilibrado do TF pelo IDF.

4.1.3 Similaridade do Cossenos

A similaridade entre cossenos é uma métrica estatística, usada em sistemas de recomendação nos campos textuais, para medir a similaridade entre os documentos. Levando em consideração um modelo de espaço vetorial e calculando o ângulo θ entre dois documentos \vec{A}

	0	1	2	3	4	5	6	7
	beat	dead	easily	evil	music	robot	rock	soul
Rock	0	0	0	0	0	0	1	0
Easily	0	0	1	0	0	0	0	0
Evil Deeds	0	1	0	1	0	0	0	0
Rock Soul	0	0	0	0	0	0	1	1
Easily	0	0	1	0	0	0	0	0
Rock BeaT	1	0	0	0	0	0	1	0
Music Rock	0	0	0	0	1	0	1	0
Robot Rock	0	0	0	0	0	1	1	0

Tabela 4.1: Valores para o TF de cada documento.

	0	1	2	3	4	5	6	7
	beat	dead	easily	evil	music	robot	rock	soul
IDF do termo	2.504	2.504	2.099	2.504	2.504	2.504	1.405	2.504

Tabela 4.2: Valores para o IDF de cada termo.

e \vec{B} (Singhal, 2001). Sendo o valor gerado da similaridade entre 0.0 e 1.0: o primeiro indica que os vetores são ortogonais entre si, enquanto o último indica que os vetores são idênticos. A Equação 4.4 é uma representação formal de como a similaridade é encontrada. Neste trabalho, o modelo de espaço vetorial é preparado pelo TF-IDF.

$$\cos(A, B) = \frac{\sum_{i=1}^n A_i \cdot B_i}{\sqrt{\sum_{i=1}^n A_i^2} \cdot \sqrt{\sum_{i=1}^n B_i^2}} \quad (4.4)$$

Por exemplo, utilizando o espaço vetorial dos resultados do TF-IDF presente na seção anterior, obtemos a Tabela 4.4 de similaridade calculando o cosseno dado a dado.

4.2 Modelagem dos dados

Os dados usados no sistema serão formalmente modelados, especificados e seus domínios definidos nesta seção.

4.2.1 Banco de dados

A modelagem física do banco de dados permite manter um controle do processo de software com maior qualidade. Inclui, portanto, uma análise das características e recursos necessários para armazenamento e manipulação das estruturas de dados (estrutura de armazenamento, endereçamento, acesso e alocação física), sendo uma sequência de comandos executados em SQL, a fim de criar as tabelas, estruturas e ligações projetadas até então e finalmente criar o

	0	1	2	3	4	5	6	7
	beat	dead	easily	evil	music	robot	rock	soul
Rock	0	0	0	0	0	0	1	0
Easily	0	0	1	0	0	0	0	0
Evil Deeds	0	0.707	0	0.707	0	0	0	0
Rock Soul	0	0	0	0	0	0	0.489	0.872
Easily	0	0	1	0	0	0	0	0
Rock BeaT	0.872	0	0	0	0	0	0.489	0
Music Rock	0	0	0	0	0.872	0	0.489	0
Robot Rock	0	0	0	0	0	0.872	0.489	0

Tabela 4.3: Valor balanceado do TF pelo IDF.

	Rock It	Easily	Evil Deeds	Rock a My Soul	Easily	Rock The BeaT	Music My Rock	Robot Rock
Rock It	1.0	0.0	0.0	0.489	0.0	0.489	0.489	0.489
Easily	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0
Evil Deeds	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Rock a My Soul	0.489	0.0	0.0	1.0	0.0	0.239	0.239	0.239
Easily	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0
Rock The BeaT	0.489	0.0	0.0	0.239	0.0	1.0	0.239	0.239
Music My Rock	0.489	0.0	0.0	0.239	0.0	0.239	1.0	0.239
Robot Rock	0.489	0.0	0.0	0.239	0.0	0.239	0.239	1.0

Tabela 4.4: Matriz de similaridade música por música.

banco de dados. A Figura 4.1 demonstra toda a estrutura de tabelas criadas para a aplicação persistir e gerenciar os dados. As tabelas que recebem o sufixo de “*RunTime*” são referentes ao tempo de execução de cada algoritmo do sistema e foram utilizadas para melhor construção da aplicação proposta. Sendo assim, não serão abordadas ao longo deste trabalho.

4.2.2 Usuário

A definição do que é um usuário tem grande importância neste trabalho, visto que o sistema proposto deseja encontrar recomendações de músicas para eles. O conjunto dos usuários U é formado por todos os usuários que podem receber recomendações do sistema. Assim, o usuário $u \in U$ será representado como um conjunto de metadado, composto unicamente por uma identificação, que é um valor único no sistema $\langle UID \rangle$, como é possível visualizar nos exemplos da Tabela 4.5, onde cada linha representa um usuários u . Na modelagem da Figura 4.1 a tabela *User* representa o conjunto dos usuários U . O usuário é representado apenas pela sua identificação, devido a este trabalho implementar a filtragem baseada em conteúdo, tornando assim desnecessário o uso de dados complementares sobre os usuários. A Equação 4.5 mostra formalmente a construção do conjunto de usuários.

$$U = \{u_1, \dots, u_n / 1 \leq n \leq N^*\} \quad (4.5)$$

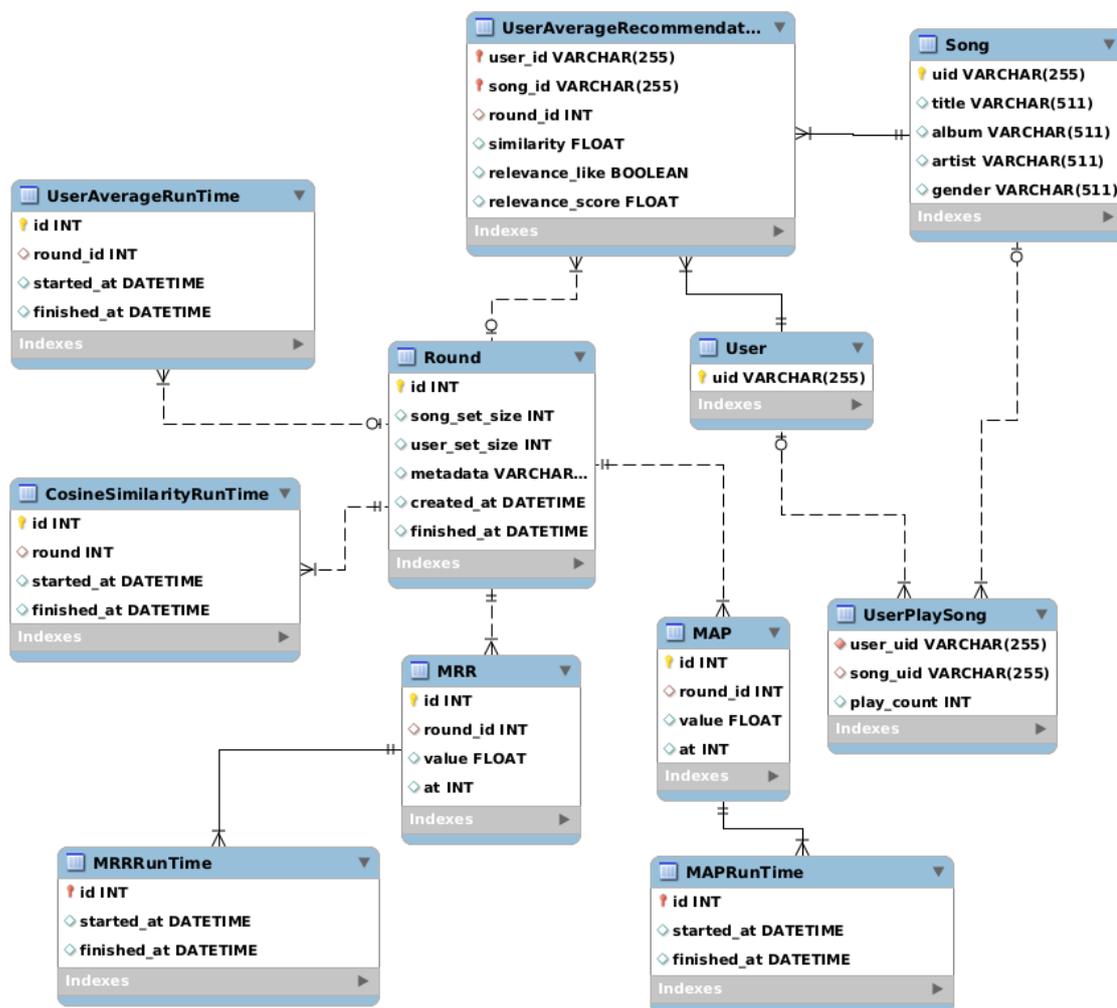


Figura 4.1: Modelagem física do banco de dados do sistema, mostrando as tabelas e as colunas.

UID
U-001
U-002
U-003
U-004
U-005

Tabela 4.5: Metadado < UID > representando 5 usuários, cada usuário em uma linha.

UID	Título	Álbum	Artista
M-001	Rock A My Soul	Demos	Pixies
M-002	Easily	Californication	Red Hot Chili Peppers
M-003	Robot Rock	Musique Vol 1	Daft Punk
M-004	Evil Deeds	Encore	Eminem
M-005	Easily	Starlight	Muse
M-006	Rock It	Rock It	Sub Focus
M-007	Rock The BeaT	Party Rock	LMFAO
M-008	Music My Rock	Bedouin Soundclash	Sounding Aмосaic

Tabela 4.6: As músicas e seus metadados.

4.2.3 Item de Recomendação: Música

A definição do que é uma música e seus metadados tem grande importância neste trabalho, visto que o sistema proposto deseja recomendar estas canções aos usuários. O conjunto das músicas M é formado por todas as músicas que podem ser recomendadas. Assim para analisar os metadados das faixas e fazer um bom uso deles durante a recomendação, formalmente cada música $m \in M$ será representada como um conjunto de metadados distinguidas em uma quintupla $\langle UID, TL, AL, AR \rangle$, que significa da esquerda pra direita as informações sobre a música: UID é a identificação única daquela canção, TL é o metadado textual que representa o Título da faixa, AL é o metadado textual que representa o Álbum ao qual esta pertence e AR é o metadado textual que representa o Artista ou Banda intérprete. Sendo possível visualizar exemplos na Tabela 4.6, onde cada linha representa uma música m . Na modelagem da Figura 4.1 a tabela *Song* representa o conjunto das músicas M . A Equação 4.6 mostra formalmente a construção do conjunto de músicas. Na Figura 4.2 é possível verificar um destes, que servirá de exemplo para esta seção de modelagem dos dados.

$$M = \{m_1, \dots, m_n / 1 \leq n \leq N^*\} \quad (4.6)$$

4.2.4 Modelo do usuário

Em busca de analisar os metadados das composições e fazer um bom uso deles para recomendar. Formalmente, o modelo do usuário C_u é representado como um conjunto de metadados das músicas $m \in M$ e do usuário $u \in U$. A Equação 4.7 mostra categoricamente a construção do modelo para todos os usuários. O modelo de um usuário C_u é composto de músicas que foram adicionadas como preferidas pelo usuário u como visto na Figura 4.3. Na modelagem da Figura 4.1 a tabela *UserPlaySong* armazena os modelos de todos os usuários, a Tabela 4.7 contém exemplos da estrutura de armazenamento, utilizando assim identificações da música e do usuário para criar o modelo.

$$(\forall u \in U) \quad C_u = \{m_{1u}, \dots, m_{nu} / m_n \in M, 1 \leq n \leq |M|\} \quad (4.7)$$

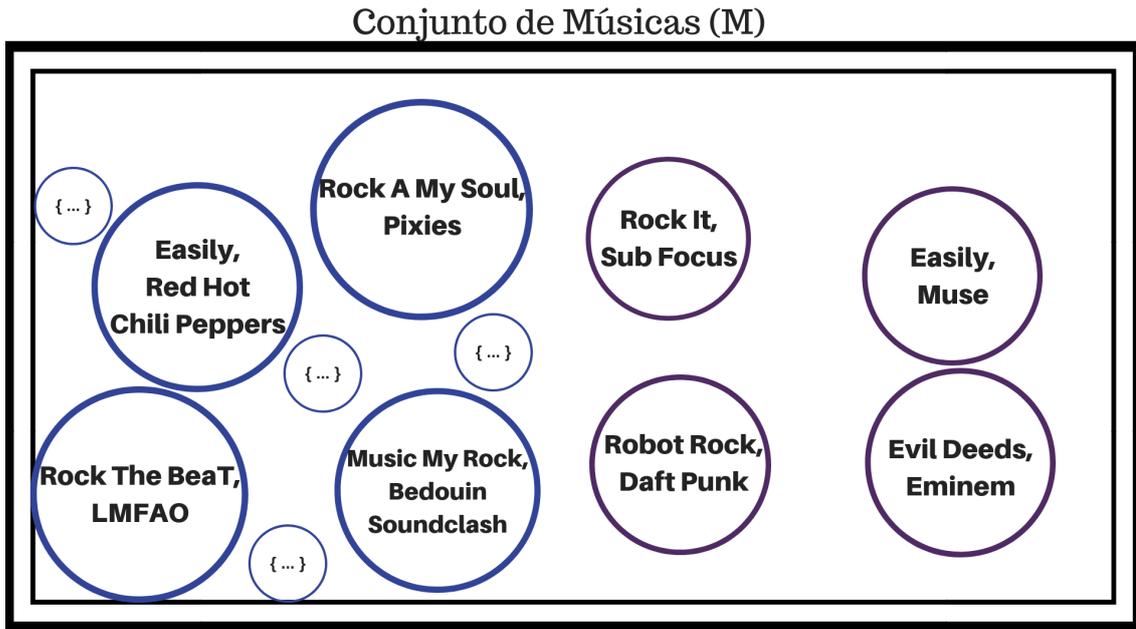


Figura 4.2: Conjunto das músicas composto por seus metadados.

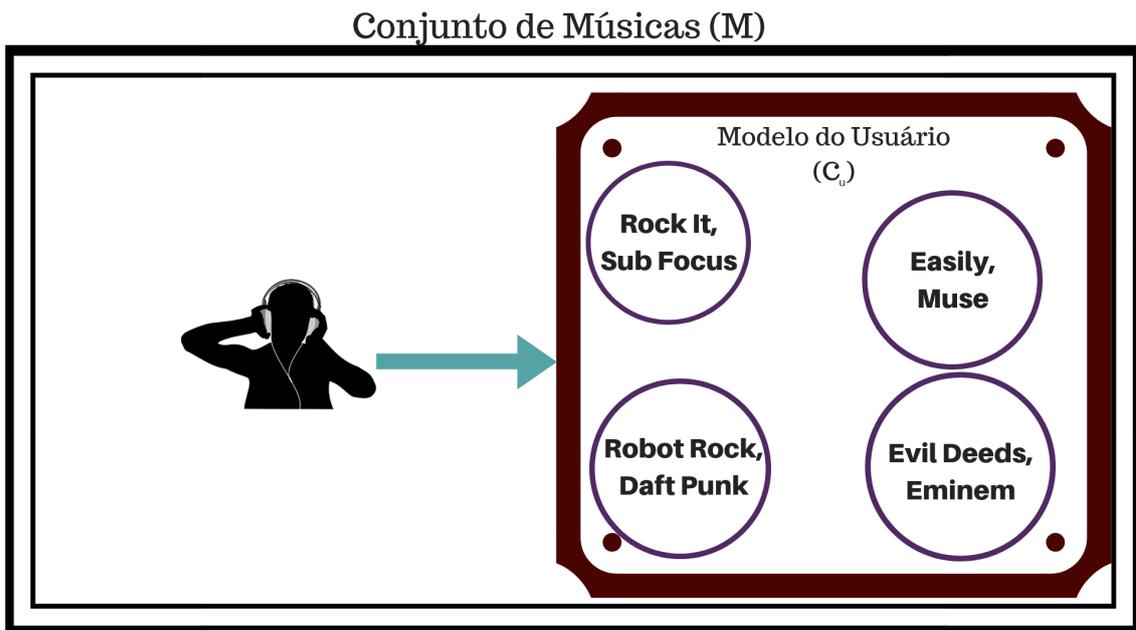


Figura 4.3: Modelo do usuário com 4 músicas adicionadas na sua preferência.

4.2.5 Modelo da Música

O modelo da música D_u é composta por todas as músicas $m \in M$ que não foram adicionadas como preferidas pelo usuário u . Ou seja, para cada $u \in U$, o conjunto das músicas é dividido em duas partes: o modelo do usuário C_u e o modelo da música D_u . A Equação 4.8 apresenta a descrição formal do modelo das músicas para todos os usuários. Na Figura 4.4 é possível verificar um exemplo de modelo das músicas, assim como a divisão do conjunto das músicas em C_u e D_u .

USER UID	SONG UID
U-001	M-003
U-001	M-004
U-002	M-005
U-002	M-006
U-003	M-003
U-003	M-004
U-003	M-005
U-003	M-006

Tabela 4.7: Modelos de 3 usuários diferentes.

$$(\forall u \in U) \quad D_u = M - C_u \quad (4.8)$$

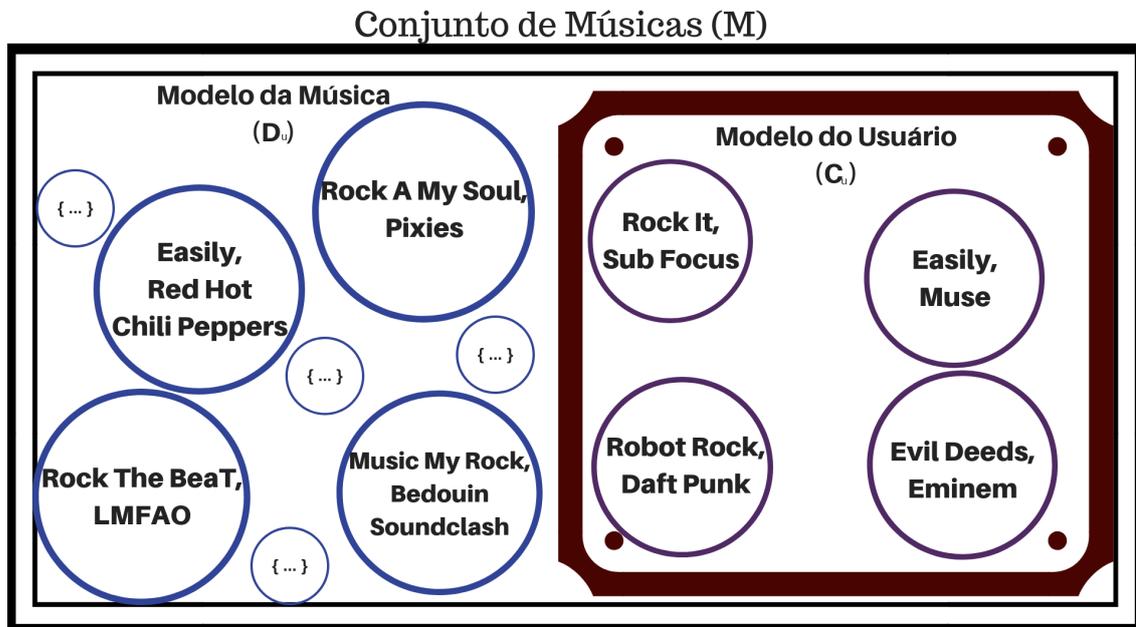


Figura 4.4: Conjunto das Músicas, Modelo da Música e Modelo do Usuário.

4.2.6 Cálculo médio do interesse dos usuários sobre uma música

A similaridade entre uma música m e um modelo do usuário C_u é dado através de um cálculo que realiza a soma dos valores das similaridades de m com cada música $A \in C_u$ e por fim realiza uma divisão pelo número de obras similares Q_m . Este cálculo é formalmente apresentado na Equação 4.9.

$$S(C_u, m) = \frac{\sum_{A=1}^{|C_u|} \cos(C_u(A), m)}{Q_m} \quad (4.9)$$

Na Tabela 4.8, por exemplo, é possível verificar que a faixa “Rock a My Soul” tem similaridade com duas outras, do modelo desse usuário: “Rock It” com similaridade 0.489

e “Robot Rock” com similaridade 0.239. Assim, somando a similaridade e dividindo por duas ocorrências temos a média do valor 0.364. Na Tabela 4.9 também é possível verificar a similaridade da música “Easily” com o conjunto.

	Rock It	Easily	Evil Deeds	Robot Rock	média
Rock a My Soul	0.489	0.0	0.0	0.239	0.364

Tabela 4.8: Similaridade da música “*Rock a My Soul*” com o D_u sendo duas músicas similares.

	Rock It	Easily	Evil Deeds	Robot Rock	média
Easily	0.0	1	0.0	0.0	1

Tabela 4.9: Similaridade da música “*Easily*” com o D_u sendo similar a uma música.

4.2.7 Modelo de Recomendação

O modelo de recomendação visa sugerir novas músicas que os usuários não conheçam, vindo assim a serem adicionadas como preferidas. A Equação 4.10 é uma definição formal deste. Para cada usuário $u \in U$, o sistema deseja recomendar músicas desconhecidas ($m^{max,u} \in D_u$) que sejam maximizadas pela função $recModel$, descrita na Equação 4.11. O $\arg \max_{m \in D_u}$ ranqueia de forma decrescente os resultados providos do cálculo médio do interesse do usuário sobre as músicas, criando assim a lista de recomendação para cada usuário.

$$\forall u \in U, m^{max,u} = \arg \max_{m \in D_u} recModel(u, m) \quad (4.10)$$

Onde o D_u são as músicas não conhecidas pelo usuário $u \in U$. E o $recModel$ é a função de similaridade entre o modelo do usuário C_u e cada música modelo de música, assim como descrito na Equação 4.11 e dada como algoritmo na Seção 4.2.6.

$$recModel(u, m) = S(C_u, m) \quad (4.11)$$

É possível observar, por exemplo, na Tabela 4.10 as composições que seriam sugeridas pelo modelo de recomendação, utilizando o exemplo da Figura 4.5.

	Rock It	Easily	Evil Deeds	Robot Rock	Similaridade
Easily	0.0	1.0	0.0	0.0	1.0
Rock a My Soul	0.489	0.0	0.0	0.239	0.364
Rock the BeaT	0.489	0.0	0.0	0.239	0.364
Music My Rock	0.489	0.0	0.0	0.239	0.364

Tabela 4.10: Lista encontrada pelo modelo de recomendação.

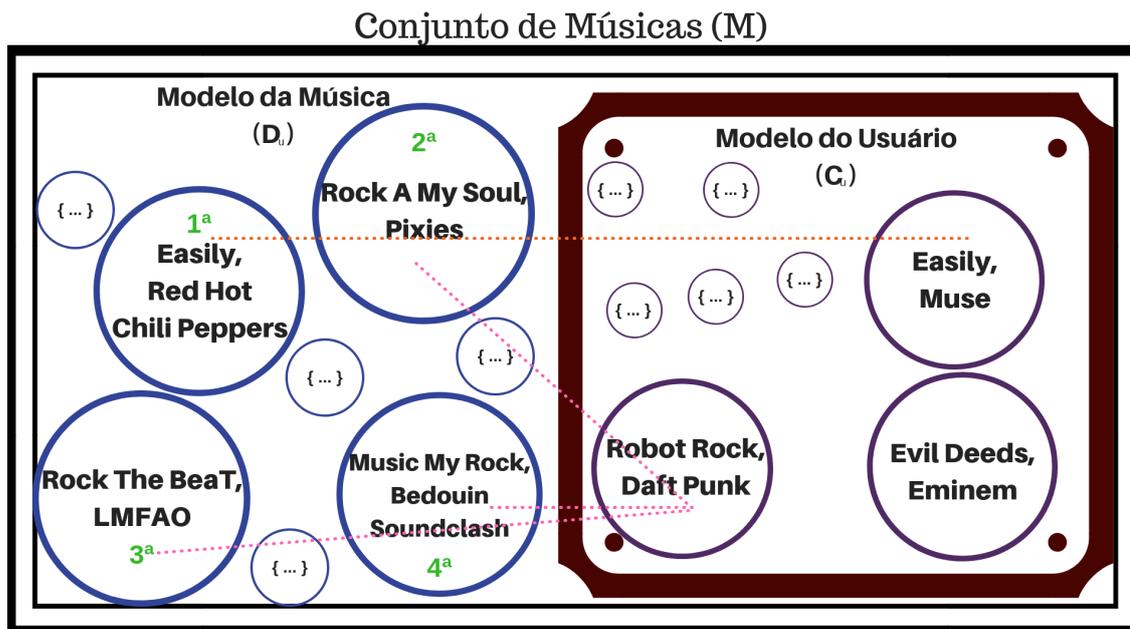


Figura 4.5: Modelo de recomendação com as músicas similares entre D_u e C_u .

4.3 Etapas de Execução

O sistema proposto foi construído para executar etapas, desde tratar o conjunto de dados escolhido até o momento de medir a assertividade das recomendações. Abaixo estão descritas todas as etapas de funcionamento do sistema, assim como visto na Figura 4.6.

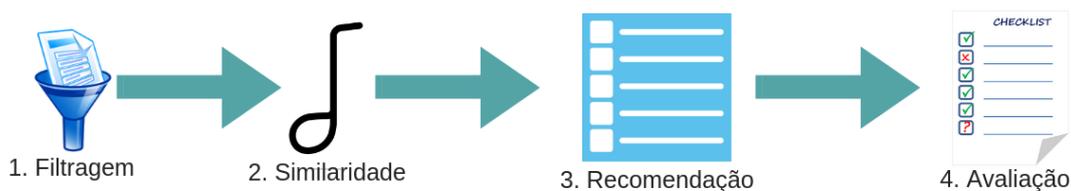


Figura 4.6: Etapas de execução das atividades do sistema para gerar uma recomendação.

- **Filtragem:** Esta primeira etapa é responsável por tratar o conjunto de dados escolhido com verificação de consistência dos dados, escolha dos metadados que são usados no sistema e organização dos dados.
- **Similaridade:** Nesta etapa os metadados passam pelo Processamento de Linguagem Natural (PLN) assim como pelo TF-IDF, que prepara o espaço vetorial a ser usado para encontrar a similaridade das músicas através do cosseno.

- **Recomendação:** Nesta etapa as recomendações são geradas a partir das definições de seus modelos com o uso da média do usuário.
- **Avaliação:** Nesta etapa, as métricas serão aplicadas para saber o quão assertivas as sugestões foram. Esta etapa será melhor descrita no capítulo seguinte.

4.4 Tecnologias

O desenvolvimento do sistema envolveu diversas tecnologias como: linguagens, bibliotecas, banco de dados, entre outros. A seguir as tecnologias usadas serão apresentadas assim como suas descrições.

4.4.1 Python

Python¹ é uma linguagem de propósito geral de alto nível, multi paradigma, suporta o paradigma orientado a objetos, imperativo, funcional e procedural. Possui tipagem dinâmica e uma de suas principais características é permitir a fácil leitura e exigir poucas linhas de código, se comparado ao mesmo programa em outras linguagens. Foi lançado por Guido van Rossum, em 1991, e atualmente possui um modelo de desenvolvimento de código aberto e gerenciado pela organização sem fins lucrativos Python Software Foundation. A linguagem foi projetada com a filosofia de enfatizar a importância do esforço do programador sobre o esforço computacional. Prioriza a legibilidade do código sobre a velocidade ou expressividade. Combina uma sintaxe concisa e clara com os recursos poderosos de sua biblioteca padrão e por módulos e *frameworks* desenvolvidos por terceiros. Para gerenciar as bibliotecas Python provê de duas ferramentas: o Ambiente Virtual² e o Pip³. O primeiro tem, como principal objetivo, criar um ambiente isolado para projetos em Python. Isso significa que cada projeto pode ter suas próprias dependências, independentemente de quais dependências cada outro projeto possui. Já o segundo é um sistema de gerenciamento usado para instalar e gerenciar as bibliotecas do Python. Durante o desenvolvimento do sistema diversas bibliotecas foram usadas, a seguir estas serão descritas:

- **Django**⁴: É um poderoso *framework* Web escrito em Python que tem chamado muita atenção da comunidade por sua facilidade de implementação, sua organização de código e suas funcionalidades. O Django pode ser utilizado em qualquer cenário de desenvolvimento web, uma vez que se trata de um *framework* bastante robusto e que suporta o desenvolvimento dos mais variados tipos de projeto, desde sites simples até grandes aplicações.

¹<https://www.python.org/>

²<https://virtualenv.pypa.io/en/stable/>

³<https://pypi.org/project/pip/>

⁴<http://www.djangoproject.com/>

- **Matplotlib**⁵: É uma biblioteca com recursos para a geração de gráficos 2D a partir de vetores. Gráficos comuns podem ser criados com alta qualidade a partir de comandos simples, inspirados nos comandos gráficos do MATLAB.
- **NLTK**⁶: É uma biblioteca para criar programas em Python para trabalhar com processamento dados em linguagem humana. Ele fornece interfaces fáceis de usar para mais de 50 recursos corpora e lexicais, juntamente com um conjunto de bibliotecas de processamento de texto para classificação, *tokenization*, derivação, marcação, análise e raciocínio semântico.
- **NumPy**⁷: É um biblioteca para a linguagem Python que suporta arrays e matrizes multidimensionais, possuindo uma larga coleção de funções matemáticas para trabalhar com estas estruturas.
- **Pandas**⁸: É uma biblioteca código aberto amplamente utilizada na comunidade acadêmica. Esta se tornou extremamente útil pelo seu desempenho e pela sua capacidade de simplificar tarefas complicadas de manipulação de dados.
- **Scikit-learn**⁹: É uma biblioteca de aprendizado de máquina de código aberto para a linguagem de programação Python. Ela inclui vários algoritmos de classificação, regressão e agrupamento, e é projetada para interagir com as bibliotecas Python numéricas e científicas NumPy e SciPy.
- **SciPy**¹⁰: É uma biblioteca de código aberto que foi feita para matemáticos, cientistas e engenheiros. Também tem o nome de uma popular conferência de programação científica com Python.

4.4.2 PostgreSQL

O PostgreSQL¹¹ é um sistema gerenciador de banco de dados (SGBD), objeto relacional de código aberto que utiliza e estende o SQL (Structured Query Language) combinando-o com diversas funcionalidades que armazenam e dimensionam com segurança as cargas de trabalho de dados. Os dados são fatos que podem ser gravados e que possuem um significado implícito. Um banco de dados representa alguns aspectos do mundo real, sendo chamado, às vezes, de minimundo ou de universo de discurso. As origens do PostgreSQL datam de 1986, como parte do projeto POSTGRES da Universidade da Califórnia em Berkeley. Um sistema gerenciador de banco de dados (SGBD) é uma coleção de programas que permite aos usuários criar e manter

⁵<https://matplotlib.org/>

⁶<http://www.nltk.org/>

⁷<http://www.numpy.org/>

⁸<https://pandas.pydata.org/>

⁹<http://scikit-learn.org/stable/>

¹⁰<https://www.scipy.org/>

¹¹<https://www.postgresql.org/about/>

um banco de dados. O PostgreSQL é executado em todos os principais sistemas operacionais, está em conformidade com o ACID atômica, preservação de consistência, isolamento e durabilidade (ou persistência).

4.5 Sumário

Neste capítulo foi apresentada uma visão geral sobre os aspectos do desenvolvimento do sistema e as tecnologias usadas. A base teórica dos algoritmos utilizados com exemplos e os dados com suas modelagens. Por fim, descreveram-se as etapas de atividades que o sistema executa para gerar a lista de recomendação. No Capítulo 5 a etapa de avaliação será descrita, assim como as métricas utilizadas e os resultados obtidos.

5

Avaliação

Pois não é toda palavra que se encaixa num bom samba

—EMICIDA (Hino Vira-Lata)

Neste capítulo, será mostrado o processo de avaliação utilizado para verificar se os objetivos previstos foram alcançados. Espera-se que em um cenário com escassez de metadados o sistema proposto consiga recomendar novas composições aos usuários, de forma personalizada. Também apresenta os detalhes da metodologia utilizada para desenvolver e avaliar este trabalho, bem como o conjunto de dados utilizados durante os experimentos. Em seguida, são apresentadas as métricas utilizadas na avaliação e os resultados obtidos. Por fim, é feita uma discussão, assim como apresentados pontos de melhoria.

5.1 Metodologia

Os testes realizados têm por objetivo mostrar que é possível recomendar canções em um cenário com escassez de informação e qual a melhor forma de utilizar os metadados. Abaixo estão descritos os três experimentos realizados:

1. **Experimento 1:** Este é composto de testes do sistema realizados de forma isolada para encontrar assim as recomendações. Supondo o uso de 3 metadados: $MT1$, $MT2$ e $MT3$. Primeiramente, será utilizado o metadado $MT1$ para encontrar as recomendações, depois o $MT2$ será usado de forma isolada para encontrar as sugestões e, por fim, o metadado $MT3$ será usado separadamente também.
2. **Experimento 2:** Os dois metadados que encontraram as melhores recomendações durante o Experimento 1 serão utilizados em conjunto para encontrar as recomendações. Ou seja, é feita uma média da similaridade dos metadados. Por exemplo, o $MT1$ e o $MT2$ encontraram melhores recomendações. Assim, a similaridade de música para música será a média das similaridades $|MT1| + |MT2|$.

3. **Experimento 3:** Os dois metadados que encontraram as melhores recomendações durante o Experimento 1 são concatenados e utilizados para encontrar as sugestões. Por exemplo, o *MT1* e o *MT2* encontraram melhores propostas, sendo assim os mesmos serão concatenados em *MT1 + MT2*.

Para avaliar os experimentos no sistema proposto e seu modelo de recomendação foram utilizados três metadados: *Título*, *Álbum* e *Artista*. As listas de sugestões encontradas em cada experimento foram avaliadas em 5 tamanhos: 1, 5, 10, 15 e 20 músicas recomendadas. O conjunto de composições relevantes ou esperadas a serem propostas aos usuários é determinado pelo desvio padrão das vezes em que a música foi adicionada nas preferências dos usuários. As métricas utilizadas para avaliar a assertividade do sistema foram: *Mean Average Precision (MAP)* e *Mean Reciprocal Rank (MRR)*.

5.2 Conjunto de dados

Todos os experimentos realizados utilizaram os dados providos pelo projeto *One Million Songs* apresentado na Seção 3.4.1.1. O conjunto de dados utilizado neste trabalho é um subconjunto, provido pelo projeto, para que estudos como este possam ser realizados. Na Tabela 5.1 encontra-se o levantamento dos dados utilizados com 10000 músicas e 76353 usuários. Os valores da média, mediana e desvio padrão representam o número de vezes que uma canção foi adicionada nas preferências dos usuários. Para este trabalho foram utilizadas 2 milhões.

	Total	Média	Mediana	Desvio Padrão	Maior que o DP	Menor que DP
Músicas	10000	200	124	317	1157	8843
Usuários	76353	26	16	31	18793	57560

Tabela 5.1: Informações sobre o conjunto de dados usado.

No histograma normalizado da Figura 5.1 é possível verificar a distribuição das composições. O valor de normalização é dado pela música que foi mais adicionada nas preferências dos usuários, sendo 8277 o total de usuários que a adicionaram e 0,038 o desvio padrão normalizado. No histograma da Figura 5.2 é possível visualizar a distribuição dos usuários, sendo o valor de normalização dado pelo perfil que mais adicionou canções em sua lista de preferência, com um valor de 711 e o desvio padrão normalizado de 0,044. Assim, é possível obter as músicas que farão parte do conjunto de músicas esperadas e também é possível obter os usuários mais ativos para encontrar indicações.

Na Tabela 5.1 apresentada acima, a coluna “Maior que o DP”, representa a quantidade de obras que fazem parte do conjunto de músicas esperadas a constituírem as listas de sugestões com 1157 e de usuários para encontrar recomendações com 18793. A coluna “Menor que o DP”, representa o oposto, ou seja, as músicas não esperadas na constituição das listas de recomendações, com 8843 músicas e 57560 usuários com poucas músicas adicionadas a preferências.

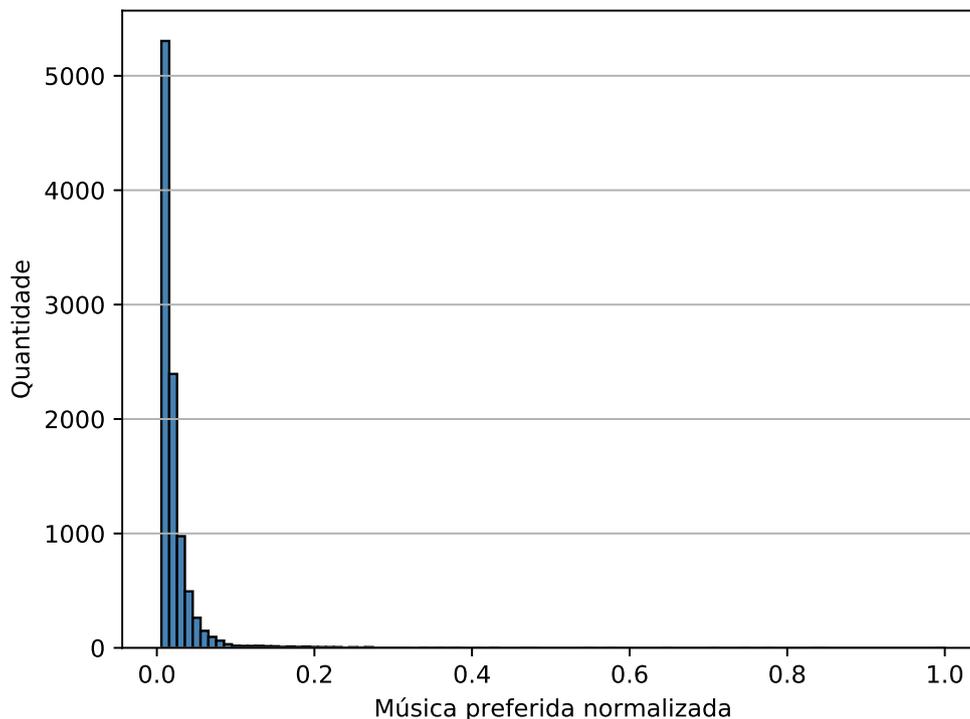


Figura 5.1: Histograma normalizado das músicas.

Com essa análise, criamos um cenário no qual 11% das músicas são consideradas relevantes e, portanto, nos habilita a realizar o cálculo das métricas. O uso do Desvio Padrão é uma decisão de projeto.

5.3 Métricas de Avaliação

Existem diversas métricas que podem ser utilizadas para avaliar a precisão de um sistema de recomendação. Neste trabalho, as métricas utilizadas foram: *MAP* e *MRR*. Assim, é considerado um acerto quando a música recomendada faz parte das 1157 “Maior que o DP”, caso contrario será considerado como uma sugestão errada. Caso a lista de recomendação não possua nenhuma canção que faça parte do conjunto de músicas esperadas, o valor do resultado é zero. As definições e a formulação do cálculo destas métricas são apresentadas nas próximas seções.

5.3.1 Mean Average Precision - MAP

A fim de obter uma métrica única que contribui para a precisão do método de recomendação ao longo de todo o conjunto de usuários, utiliza-se o *MAP* (Parra and Sahebi, 2013). O valor do *MAP* é obtido calculando a média sobre a precisão média da lista de recomendações de cada usuário.

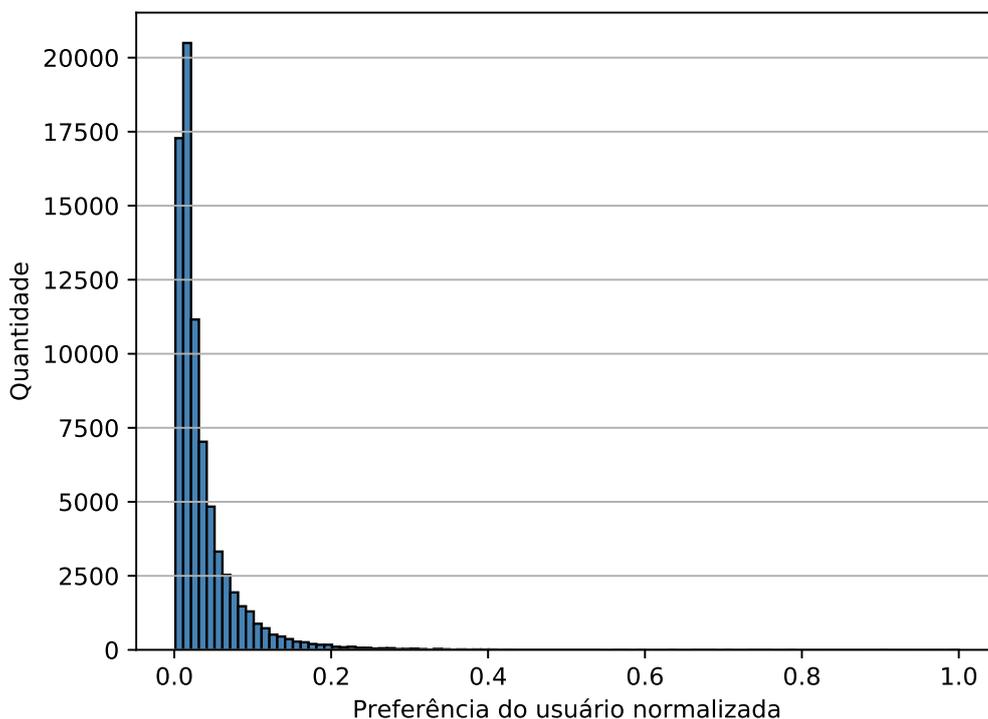


Figura 5.2: Histograma normalizado dos usuários.

$$MAP = \frac{1}{|U|} \sum_u^{U} AveP(u) \quad (5.1)$$

$$AveP(u) = \frac{1}{N} \sum_{i=1}^N p@i \quad (5.2)$$

$$p@i = \frac{r}{i} \quad (5.3)$$

Na Equação 5.1, $AveP(u)$ é a média da precisão para o usuário $u \in U$, isto é, a média dos valores da precisão obtidos para o conjunto de top-N recomendações depois que cada sugestão relevante é recuperada (Manning *et al.*, 2008). A Equação 5.2 e 5.3 mostra o cálculo da média da precisão, que é uma soma da precisão em cada posição da lista $p@i$ onde r é a quantidade de músicas relevantes até a posição i . Na Tabela 5.2 encontram-se as colunas com o cálculo da posição $p@i$ para 3 usuários diferentes, assim como a média da precisão para os mesmo e por fim o valor do MAP dado como 0,497.

	p@1	p@2	p@3	p@4	p@5	p@6	p@7	p@8	p@9	p@10	Média da Precisão
u_1	1/1	1/2	1/3	2/4	2/5	3/6	4/7	5/8	5/9	5/10	0,548
u_2	0	0	1/3	2/4	3/5	3/6	3/7	3/8	4/9	4/10	0,358
u_3	0	1/2	2/3	3/4	4/5	4/6	4/7	5/8	6/9	6/10	0,584
											0,497

Tabela 5.2: Exemplo do cálculo do MAP para tamanho 10 da lista.

u_1	u_2	u_3	MRR
1/1	1/3	1/2	0,611

Tabela 5.3: Exemplo do cálculo do MRR para três usuários.

5.3.2 Mean Reciprocal Rank - MRR

O **MRR** avalia a qualidade das listas de recomendação na posição top-N, medindo o quão longe o primeiro item relevante está do topo da lista (Qin, 2013). É definido como:

$$MRR = \frac{1}{|U|} \sum_u \frac{1}{p_i} \quad (5.4)$$

Na Equação 5.4 é possível verificar a formulação do **MRR** onde $\forall u \in U$ a primeira ocorrência de uma recomendação relevante p_i é avaliada. Utilizando o exemplo da Tabela 5.2 é possível verificar que para o usuário u_1 a primeira ocorrência da música relevante é na primeira posição $p@1$, assim o valor é $\frac{1}{1}$. Já para o u_2 a ocorrência acontece na terceira posição $p@3$, assim o valor é $\frac{1}{3}$. Na Tabela 5.3 é possível visualizar os valores do **RR** de cada usuário assim como o valor 0,611 do **MRR**.

5.4 Resultados

Para avaliar os resultados, 6 casos de testes foram aplicados. Ao qual 3 com 1000 usuários e variações de 3000, 6000 e 9000 músicas. E os outros 3 testes serão com 10000 canções e variações de 100, 500 e 1000 usuários. Todos os testes utilizaram das métricas **MAP** e **MRR**.

5.4.1 MAP

A métrica **MAP** foi utilizada para medir a assertividade do sistema perante todos os seus usuários e aplicada em todos os experimentos. Nas Figuras 5.3, 5.4, 5.5 e 5.6 encontram-se os resultados dos testes.

Os testes realizados com 10000 composições e as variações no número de usuários apresentam, nas Figuras 5.3 e 5.4, que, no sistema proposto, os Experimentos 2 e 3 obtiveram os melhores resultados. Também é possível visualizar, na Figura 5.3, que quanto maior a lista de recomendação os resultados vão sofrendo variações. O Experimento 1 em todos os testes

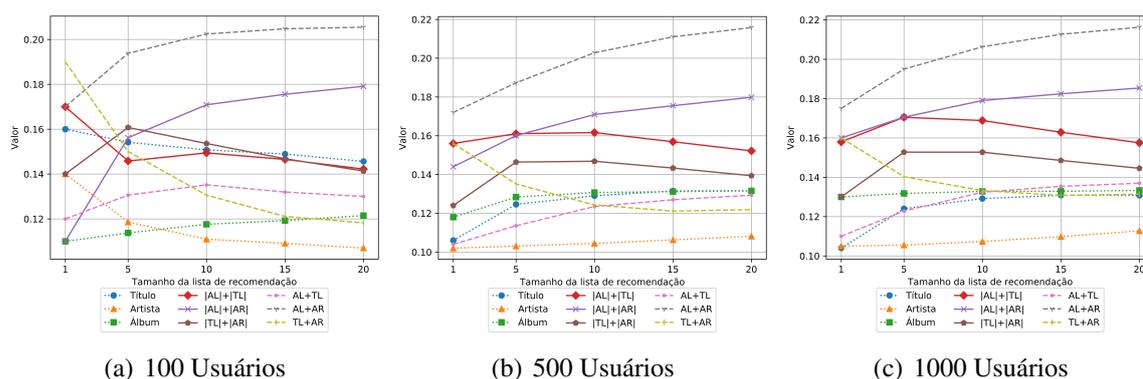


Figura 5.3: MAP - Gráfico com linhas dos experimentos com 10000 músicas e três variações no número de usuários: 100, 500 e 1000.

obteve os piores resultados, sendo o metadado *Artista* o com a pior precisão a ser trabalhado de forma isolada. O Experimento 2 obteve bons resultados quando se olha para todas as médias da similaridade dos metadados. O Experimento 3, no sistema proposto, obteve melhores resultados com a concatenação dos metadados *AL+AR* (*Álbum* e *Artista*), entretanto, as outras concatenações de metadados deste experimento, obtiveram resultados menores do que as Experimento 2. A Figura 5.4 ressalta em sua análise que todos os experimentos obtiveram *outliers*, sendo a maioria com um *outlier* e unicamente a concatenação dos metadados *AL+TL* (*Álbum* e *Título*) da Figura 5.4(a) obteve dois *outliers*. Com exceção dos *outliers*, a variação dos resultados é pequena, exceto o *TL+AR* (*Título* e *Artista*) na Figura 5.4(a).

Os testes realizados com 1000 usuários e as variações no número de obras apresentam, nas Figuras 5.5 e 5.6, que no sistema proposto o Experimento 3 obteve melhores resultados quanto maior o número de composições. O Experimento 1 obteve os piores resultados, sendo o metadado *Artista* o com a pior precisão a ser trabalhado de forma isolada. O Experimento 2 obteve bons resultados principalmente no teste da Figura 5.5(a), onde duas médias das similaridades obtiveram os melhores resultados $|TL|+|AR|$ (*Título* e *Artista*) e $|AL|+|TL|$ (*Álbum* e *Título*). O Experimento 3 no sistema proposto obteve bons resultados com a concatenação dos metadados *AL+AR* (*Álbum* e *Artista*) assim como é possível verificar nos testes das Figuras 5.5(b) e 5.5(c), entretanto as outras concatenações dos metadados obtiveram resultados menores que o Experimento 2. A Figura 5.6 ressalta em sua análise que todos os experimentos obtiveram *outliers*, sendo que a maioria obteve apenas um *outlier* com exceção da concatenação dos metadados *AL+TL* (*Álbum* e *Título*) e o metadado isolado *Título* assim como é possível visualizar na Figura 5.4(a) que obtiveram dois *outliers*. Com exceção dos *outliers* a variação dos resultados em sua maioria é pequena exceto $|TL|+|AR|$ (*Título* e *Artista*) e $|AL|+|TL|$ (*Álbum* e *Título*) na Figura 5.6(a).

Dentre os seis testes realizados, a combinação dos metadados *AL+AR* (*Álbum* e *Artista*) pertencente ao Experimento 3 obteve cinco dos melhores resultados. Assim como o metadado *Artista*, que trabalhado de forma isolada, obteve todos os piores resultados no sistema proposto.

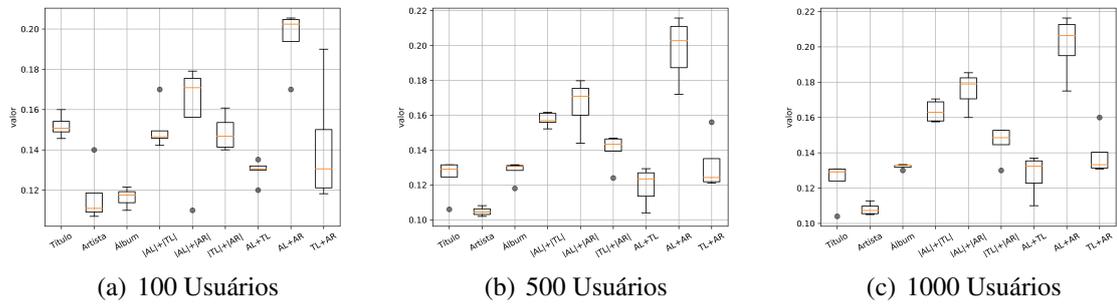


Figura 5.4: MAP - Gráfico com caixas dos experimentos com 10000 músicas e três variações no número de usuários: 100, 500 e 1000.

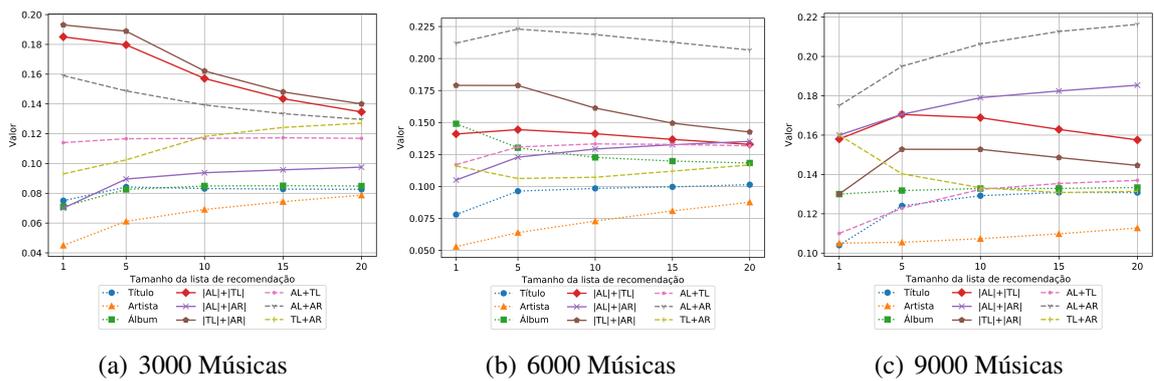


Figura 5.5: MAP - Gráfico com linhas dos experimentos com 1000 usuários e três variações no número de músicas: 3000, 6000 e 9000.

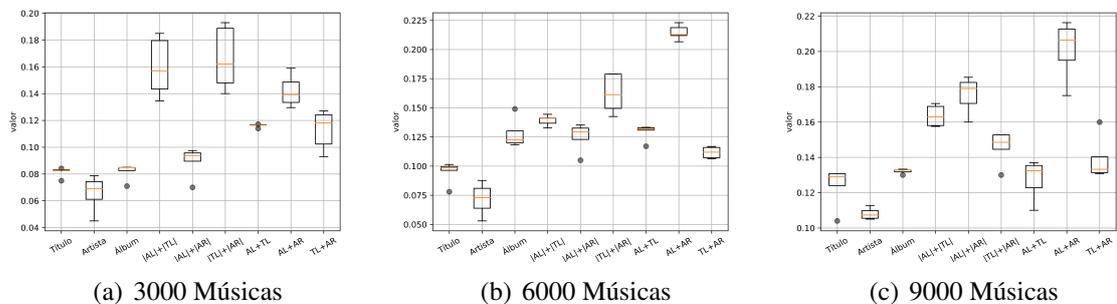


Figura 5.6: MAP - Gráfico com caixas dos experimentos com 1000 usuários e três variações no número de músicas: 3000, 6000 e 9000.

5.4.2 MRR

A métrica **MRR** foi utilizada para medir a assertividade do sistema, quanto a primeira música relevante, perante todos os seus usuários e aplicada em todos os experimentos. Nas Figuras 5.7, 5.8, 5.9 e 5.10 encontram-se os resultados dos testes.

Os testes realizados com 10000 músicas e as variações no número de usuários apresentam, nas Figuras 5.7 e 5.8, que no sistema proposto, o Experimento 2 obteve os melhores resultados. Também é possível visualizar, na Figura 5.7, que quanto maior a lista de recomendação melhor o sistema se comporta, obtendo assim melhores resultados. O Experimento 1 obteve os piores resultados, ficando sempre com algum metadado com a menor assertividade. Nas Figuras 5.7(a) e 5.8(a), entretanto, é possível visualizar que o *Título* obteve o melhor resultado no teste com 100 usuários, o que não se aplica para os demais testes. O Experimento 2, em todos os testes das Figuras 5.7 e 5.8, esteve com pelo menos uma média da similaridade dos metadados entre os três melhores resultados, sendo que a média da similaridade $|AL|+|TL|$ (*Álbum e Título*) obteve os melhores resultados nos testes das Figuras 5.7(b) e 5.7(c). O Experimento 3 no sistema proposto obteve resultados satisfatórios, porém sempre menor que o Experimento 2. A Figura 5.8 ressalta em sua análise que todos os experimentos obtiveram um *outlier* apenas, este que corresponde ao valor do **MRR** para a lista de recomendação com apenas uma canção. A variação dos resultados é aproximadamente igual, sendo que todos os experimentos obtêm melhores resultados quanto maior a lista de sugestões, obtendo assim uma curvatura de crescimento parecida, como é possível verificar na Figura 5.7.

Os testes realizados com 1000 usuários e as variações no número de obras apresentam, nas Figuras 5.9 e 5.10, que no sistema proposto, o Experimento 2 obteve dois dos três melhores resultados em todos os testes. Também é possível visualizar, na Figura 5.9, que quanto maior a lista de recomendação melhor o sistema se comporta, obtendo assim melhores resultados. O Experimento 1 obteve os piores resultados, sendo o metadado *Artista* com o a pior precisão a ser trabalhado de forma isolada. O Experimento 2 obteve os melhores resultados, com exceção do teste da Figura 5.9(b), onde a concatenação dos metadados $AL+AR$ (*Álbum e Artista*) do Experimento 3 obteve um resultado melhor seguido de duas médias das similaridades do experimento. A Figura 5.10 ressalta em sua análise que todos os experimentos obtiveram um *outlier* apenas, este que corresponde ao valor do **MRR** para a lista de recomendação com apenas uma música. A variação dos resultados é aproximadamente igual, sendo que todos os experimentos obtêm melhores resultados quanto maior a lista de sugestões, obtendo assim uma curvatura de crescimento parecida, como é possível verificar na Figura 5.9.

Dentre os seis testes realizados, a média da similaridade entre $|AL|+|TL|$ (*Álbum e Título*) pertencente ao Experimento 2 obteve quatro dos melhores resultados. Assim como o metadado *Artista* trabalhado isolado obteve cinco dos seis os piores resultados no sistema proposto.

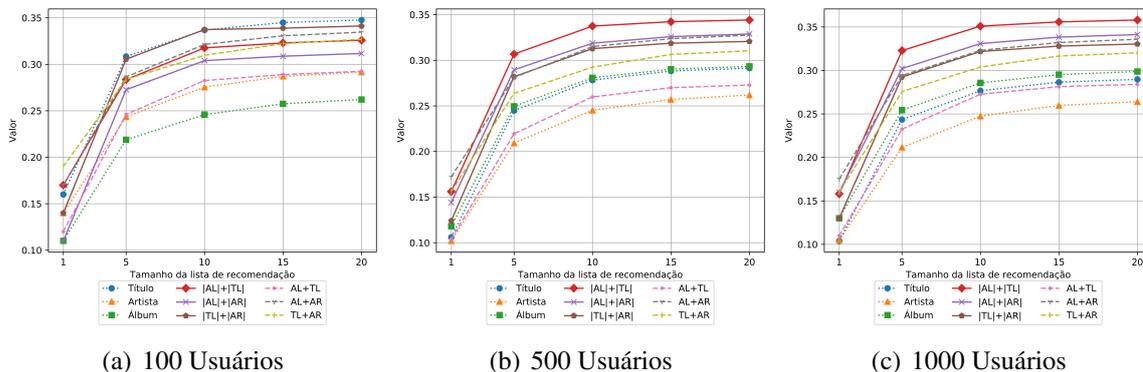


Figura 5.7: MRR - Gráfico com linhas dos experimentos com 10000 músicas e três variações no número de usuários: 100, 500 e 1000.

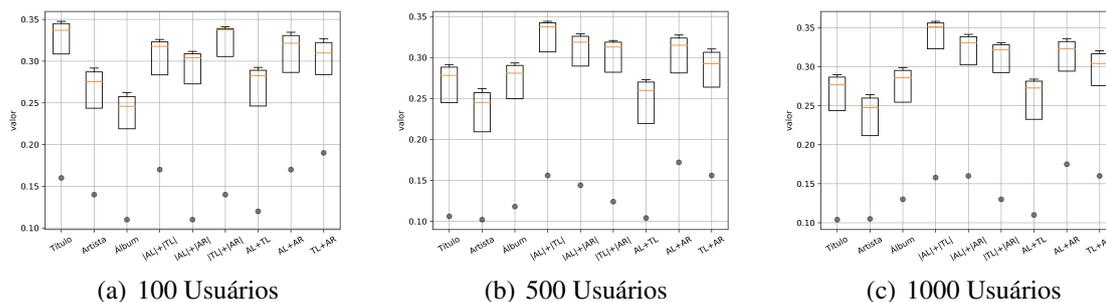


Figura 5.8: MRR - Gráfico com caixas dos experimentos com 10000 músicas e três variações no número de usuários: 100, 500 e 1000.

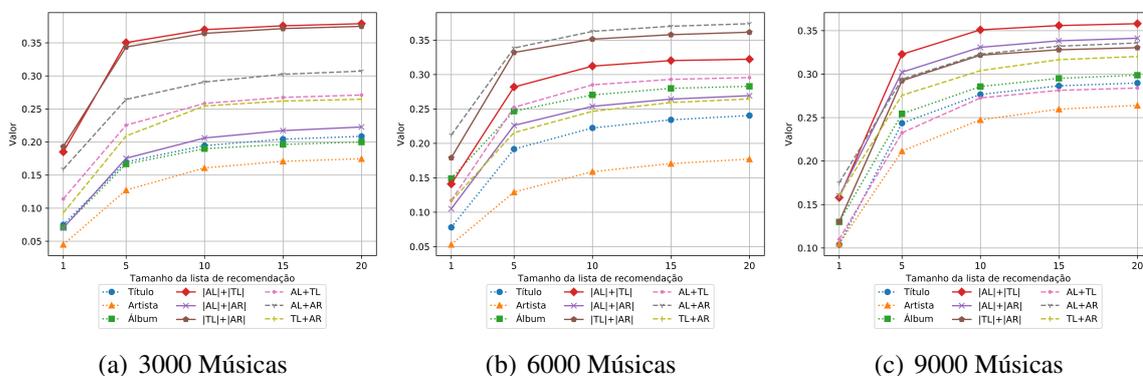


Figura 5.9: MRR - Gráfico com linhas dos experimentos com 1000 usuários e três variações no número de músicas: 3000, 6000 e 9000.

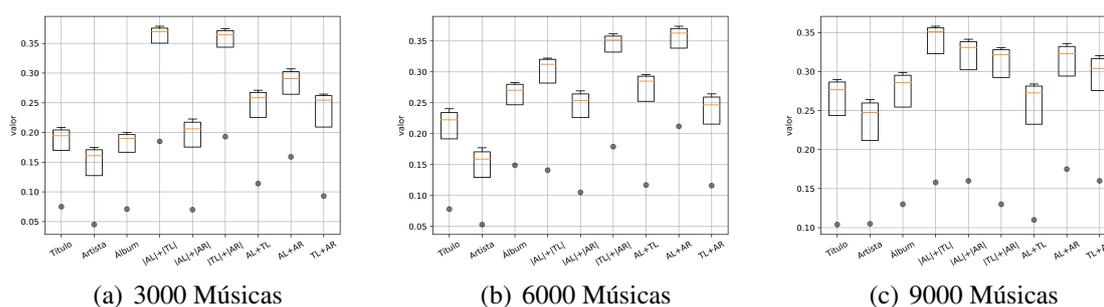


Figura 5.10: MRR - Gráfico com caixas dos experimentos com 1000 usuários e três variações no número de músicas: 3000, 6000 e 9000.

5.5 Discussão

Neste trabalho foi mostrado como construir um sistema de recomendação de música com filtragem baseado conteúdo, em um cenário com escassez de informação que descreve a música. E encontrar recomendações a partir da similaridade entre metadados textuais.

Após analisar os resultados obtidos com as métricas utilizadas, conclui-se que mesmo em um ambiente desbalanceado é possível obter aproximadamente 20% de assertividade ao longo das listas de recomendações. Quando se usa a concatenação dos metadados *AL+AR* (*Álbum* e *Artista*), o aumento no número de usuários no sistema interfere no resultado da métrica *MAP*. Não o afeta, porém, de forma brusca, assim como uma lista de recomendação com um número maior de músicas ajuda na obtenção de um melhor resultado. Já quando olhamos para a primeira canção mais relevante, a soma da similaridade *|AL|+|TL|* (*Álbum* e *Título*) comportou-se melhor, chegando a mais de 35% de assertividade. O aumento no número de músicas do sistema não interfere no resultado da métrica *MRR*, entretanto, a lista de recomendação com um número maior de músicas interfere diretamente no resultado positivo da métrica.

Os resultados obtidos são satisfatórios, levando em conta que o ambiente trabalhado possuía apenas 11% de músicas relevantes aos usuários. Assim, obter um resultado da métrica *MAP* acima de 11% mostra que o modelo de recomendação proposto consegue recomendar músicas aos usuários em um cenário de escassez de informação.

5.6 Pontos de Melhorias

O principal problema que este trabalho abordou foi o cenário com escassez de informação que descreve a música, entretanto, os metadados *Título*, *Álbum* e *Artista* não descrevem a música por completo, podendo, assim, ser complementados com a *Letra* e *Gênero*, e a ainda continuar no cenário com escassez de informação. Aqui, um novo ponto de melhoria aparece. O conjunto de dados do projeto *One Million Songs* não disponibiliza esses metadados diretamente, mas através de pesquisas póstumas, sendo necessário realizar uma ligação entre os metadados. Para isto, ter-se-ia que trabalhar com toda a grande massa de dados do projeto, para criar um subconjunto

com todas as informações que seriam usadas. Devido a isso, escolheu-se o subconjunto já provido pelo projeto para realização de trabalhos como este.

Outro problema na realização deste trabalho foi o tempo de execução dos algoritmos, como a similaridade é calculada música a música e a recomendação é encontrada a partir da similaridade entre dois modelos que contém todas as músicas do sistema, realizar esta tarefa não foi trivial para os computadores. O limite superior da complexidade de tempo do modelo de recomendação é $\mathcal{O}(|U| \times |C_u| \times |D_u|)$, onde a quantidade de usuários $|U|$ é multiplicada pelo número de música no modelo do usuário multiplicado pelo número de músicas no modelo da música.

Outro ponto de melhoria é o processo de pré-processamento dos metadados em linguagem natural da base de dados. Como as músicas são lançadas em diferentes línguas, o processo de *tokenization* e derivação precisa identificar a língua daquele metadado, para assim aplicar as tarefas de PLN. Desta forma, atualmente se aplicado apenas às regras do Inglês Norte Americano.

5.7 Sumário

Este capítulo apresentou os principais resultados obtidos, durante o desenvolvimento dos experimentos, para avaliar o sistema proposto e seus algoritmos. Foram apresentadas as metodologias de avaliação, a base de dados a as métricas utilizadas. O estudo apresentado neste capítulo mostrou que é possível construir um sistema de recomendação de música em um cenário com escassez de informação, obtendo resultados positivos com um conjunto de dados desbalanceado.

6

Conclusão

Antes de me despedir deixo ao sambista mais novo o meu pedido final

—ALCIONE (A voz do Samba)

Este trabalho de conclusão de curso apresentou um sistema de recomendação de músicas baseado em conteúdo. Inicialmente, foi apresentada a motivação para a criação do sistema de recomendação. Relatou-se o problema de recomendação de músicas em um ambiente com informações escassas. Foi proposta uma solução para ajudar aos usuários a receberem sugestões de sua preferência.

O Capítulo 2 introduz o referencial teórico da área de Sistemas de Recomendação, apresentando um histórico, discussão de conceitos quanto aos dados que são utilizados em um sistema, além das tarefas desempenhadas pelos sistemas de recomendação, as suas principais técnicas e uma comparação.

No Capítulo 3, inicialmente, apresentou-se um histórico da gravação e reprodução da música, seguido das aplicações de comércios eletrônicos que utilizam tais sistemas, como *Amazon* e *Submarino*, e serviços de transmissão de músicas como *Spotify*, *Last.fm* e *Youtube*. Em seguida mostrou-se diversos trabalhos acadêmicos relacionados a sistemas de recomendação, com foco em música. Por fim, apresentou-se APIs e Base de dados que disponibilizam informações para que trabalhos como este possam ser desenvolvidos.

O Capítulo 4 apresenta e detalha toda a proposta de solução para a recomendação de músicas, inicialmente apresentando os algoritmos usados, assim como suas formulações e exemplos. Em seguida, apresenta a modelagem de dados tanto do banco como a dos modelos usados, as etapas do sistema proposto para encontrar e medir as recomendações, e as tecnologias usadas.

Para finalizar, foi apresentada em detalhes a metodologia e realizou-se uma avaliação experimental. Os métodos de avaliação e os resultados foram apresentados, obtendo um resultado satisfatório para proposta, mostrando que é possível sugerir músicas em um ambiente com escassez de informações.

6.1 Contribuições do Trabalho

As principais contribuições deste trabalho são explicadas a seguir:

- **Revisão de sistemas de recomendação de música:** Foi realizada uma revisão de trabalhos na literatura de sistemas de recomendação de música, mostrando os trabalhos relacionados. Além disso, o estudo proporciona conhecer funcionalidades e diferentes tipos de sistemas de recomendação que poderiam ser utilizados em diferentes cenários para sugestão de músicas e criação de listas de reprodução.
- **Modelo de recomendação de músicas:** O principal foco do trabalho foi propor um modelo de recomendação de músicas, com o objetivo de recomendar músicas em um ambiente com informações escassas.
- **Avaliação Experimental:** Com o sistema de recomendação desenvolvido, foram realizados testes para avaliar o modelo de recomendação. Os testes foram realizados com diversas variações no número de músicas, usuários e os atributos da base de dados das músicas.

6.2 Trabalhos Futuros

O sistema de recomendação desenvolvido nesse projeto possui pontos de melhorias, dentre eles podemos destacar:

- **Aumento na base de dados das músicas e dos usuários:** Verificou-se que o aumento do número de usuários tem uma pequena interferência no valor da métrica, assim como quanto maior o tamanho da lista de recomendação melhor o resultado, expondo assim a necessidade de uma lista com muitas músicas. Para isso, um maior número de músicas é necessário para compor a base.
- **Uso de mais metadados do item de recomendação:** Como o sistema de recomendação proposto trabalha com os metadados das músicas, uma maior quantidade de informação sobre a mesma se faz necessário. Assim como visto nos resultados, é melhor trabalhar com metadados juntos do que de forma individual.
- **Comparação com outros algoritmos de recomendação:** A técnica de recomendação baseada em conteúdo utilizada neste trabalho possui outros algoritmos de recomendação, assim, é possível implementar estes algoritmos no sistema a fim de compará-los, para ver qual obtêm o melhor desempenho.

- **Pesos nas preferências:** Pode-se ajustar o modelo do usuário dando pesos ponderados a suas preferências a partir do número de vezes que o usuário reproduziu a música.
- **Análise de sentimento:** A partir dos comentários sobre determinadas músicas, pode-se coletar o sentimento do usuário sobre uma determinada canção e enriquecer o modelo do usuário. Esse pode ser um novo aspecto na recomendação, tornando-a mais eficaz. Assim como é possível coletar os sentimentos que constroem as composições, a partir dos metadados textuais, permitindo, assim, identificar quais são os sentimentos das músicas que pertencem ao modelo do usuário para então recomendar outras que tenham sentimentos similares.
- **Uso do áudio da música:** Além dos metadados é possível trabalhar com os atributos do áudio da música, como já visto nos capítulos de base teórica.

6.3 Sumário

Este capítulo apresentou um resumo de tudo que foi realizado e discutido neste trabalho. Mostrou-se as principais contribuições da nossa proposta de sistema de recomendação e pontos de melhorias para proposta de sistemas de recomendação de músicas a serem realizadas em trabalhos futuros.

Bibliografia

- Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering*, (6), 734–749.
- Buldú, J. M., Cano, P., Koppenberger, M., Almendral, J. A., and Boccaletti, S. (2007). The complex network of musical tastes. *New Journal of Physics*, **9**(6), 172.
- Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, **12**(4), 331–370.
- Burke, R. (2007). The adaptive web. chapter Hybrid Web Recommender Systems, pages 377–408. Springer-Verlag, Berlin, Heidelberg.
- Celma, Ò. (2014). *Music Recommendation and Discovery: The Long Tail, Long Tail, and Long Play in the Digital Music Space*. Springer Berlin Heidelberg.
- Celma, O. and Lamere, P. (2008). If you like the beatles you might like...: A tutorial on music recommendation. In *Proceedings of the 16th ACM International Conference on Multimedia, MM '08*, pages 1157–1158, New York, NY, USA. ACM.
- Celma, O. and Lamere, P. (2011). Music recommendation and discovery revisited. In *Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11*, pages 7–8, New York, NY, USA. ACM.
- Cho, S.-H., Kim, Y.-H., and Park, J.-B. (2007). Music recommendation system for public places based on sensor network. *IJCSNS*, **7**(8), 172.
- DMR (2018). 65 amazing spotify statistics and facts. Disponível em: <https://expandedramblings.com/index.php/spotify-statistics/>. Último acesso em 24 de Abril de 2018.
- Eck, D., Lamere, P., Bertin-Mahieux, T., and Green, S. (2008). Automatic generation of social tags for music recommendation. In *Advances in neural information processing systems*, pages 385–392.
- Estadão (2017). Pela primeira vez, streaming lidera receitas da música nos eua. Disponível em: <http://link.estadao.com.br/noticias/geral,pela-primeira-vez-streaming-lidera-receitas-da-musica-nos-eua,70001721714>. Último acesso em 24 de Abril de 2018.

- Fields, B. *et al.* (2011). *Contextualize your listening: the playlist as recommendation engine*. Ph.D. thesis.
- Flexer, A., Schnitzer, D., Gasser, M., and Widmer, G. (2008). Playlist generation using start and end songs. In *ISMIR*, volume 8, pages 173–178.
- Forbes (2015). One trillion songs have been streamed so far in 2015. how much larger can this get? Disponível em: <https://www.forbes.com/sites/hughmcintyre/2015/08/13/one-trillion-songs-have-been-streamed-so-far-in-2015-how-much-larger-amp/>. Último acesso em 24 de Abril de 2018.
- G1, G. (2017). Streaming de música cresce 52triplo de venda de discos. Disponível em: <https://g1.globo.com/musica/noticia/streaming-de-musica-cresce-52-no-brasil-em-2016-e-ja-rende-o-triplo-cgh.html>. Último acesso em 24 de Abril de 2018.
- Globo, O. (2012). Canção francesa gravada em 1860 é a mais antiga do mundo. Disponível em: <https://oglobo.globo.com/cultura/cancao-francesa-gravada-em-1860-a-mais-antiga-do-mundo-3619581>. Último acesso em 27 de Julho de 2018.
- Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Commun. ACM*, **35**(12), 61–70.
- Goto, M. and Goto, T. (2005). Musicream: New music playback interface for streaming, sticking, sorting, and recalling musical pieces.
- Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, **22**(1), 5–53.
- Hijkata, Y., Iwahama, K., and Nishida, S. (2006). Content-based music filtering system with editable user profile. In *Proceedings of the 2006 ACM Symposium on Applied Computing, SAC '06*, pages 1050–1057, New York, NY, USA. ACM.
- Hint, S. (2016). Sistema de recomendação da amazon e seus segredos. Disponível em: <http://www.smarthint.co/sistema-de-recomendacao-da-amazon-e-seus-segredos/>. Último acesso em 24 de Abril de 2018.
- Hornung, T., Ziegler, C. N., Franz, S., Przyjaciel-Zablocki, M., Schätzle, A., and Lausen, G. (2013). Evaluating hybrid music recommender systems. In *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 1, pages 57–64.

- IFPI (2018). Ifpi global music report 2018. Disponible in: <http://www.ifpi.org/news/IFPI-GLOBAL-MUSIC-REPORT-2018>. Last access in 28 May 2018.
- Isinkaye, F., Folajimi, Y., and Ojokoh, B. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, **16**(3), 261 – 273.
- Knees, P., Pohle, T., Schedl, M., and Widmer, G. (2006). Combining audio-based similarity with web-based data to accelerate automatic music playlist generation. In *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, MIR '06*, pages 147–154, New York, NY, USA. ACM.
- Lemire, D. and Maclachlan, A. (2005). Slope one predictors for online rating-based collaborative filtering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 471–475. SIAM.
- Logan, B. (2004). Music recommendation from song sets. Citeseer.
- Lops, P., de Gemmis, M., and Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 73–105. Springer US.
- Magno, T. (2008). A comparison of signal based music recommendation to genre labels, collaborative filtering, musicological analysis, human recommendation and random baseline.
- Mahieux, B., Ellis, D. P., Whitman, B., and Lamere, P. (2011). The million song dataset. *12th International Society for Music Information Retrieval Conference*, **2**(9), 591–596.
- Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.
- McFee, B. and Lanckriet, G. R. (2009). Heterogeneous embedding for subjective artist similarity. In *ISMIR*, pages 513–518.
- McFee, B., Mahieux, B., Ellis, D. P., and Lanckriet, G. R. (2012). The million song dataset challenge. *21st International Conference on World Wide Web*, pages 909–916.
- Mix, C. (2013). A primeira música gravada. Disponível em: musica.culturamix.com/curiosidades/a-primeira-musica-gravada. Último acesso em 27 de Julho de 2018.
- Mladenic, D. (1999). Text-learning and related intelligent agents: A survey. *IEEE Intelligent Systems*, **14**(4), 44–54.
- News, D. M. (2018). What streaming music services pay. Disponível em: <https://www.digitalmusicnews.com/2018/01/16/>

- [streaming-music-services-pay-2018/](#). Último acesso em 24 de Abril de 2018.
- Pachet, F. (2003). Content management for electronic music distribution. *Commun. ACM*, **46**(4), 71–75.
- Pampalk, E. and Goto, M. (2006). Musicrainbow: A new user interface to discover artists using audio-based similarity and web-based labeling. Citeseer.
- Pampalk, E., Pohle, T., and Widmer, G. (2005). Dynamic playlist generation based on skipping behavior.
- Parra, D. and Sahebi, S. (2013). Recommender systems: Sources of knowledge and evaluation metrics. In J. D. Velásquez, V. Palade, and L. C. Jain, editors, *Advanced Techniques in Web Intelligence-2*, volume 452 of *Studies in Computational Intelligence*, pages 149–175. Springer Berlin Heidelberg.
- Pichl, M., Zangerle, E., and Specht, G. (2017). Improving context-aware music recommender systems: Beyond the pre-filtering approach. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, ICMR '17*, pages 201–208, New York, NY, USA. ACM.
- Qin, Y. (2013). *A Historical Survey of Music Recommendation Systems: Towards Evaluation*. Ph.D. thesis.
- Resnick, P. and Varian, H. R. (1997). Recommender systems. *Commun. ACM*, **40**(3), 56–58.
- Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors (2011). *Recommender Systems Handbook*. Springer.
- Roy, T. F. (2000). The representational state transfer (rest). *Department of Information and Computer Science, UCI*.
- Schafer, B., J. B., Frankowski, D., , D., , H., , J., , S., and Sen, S. (2007). Collaborative filtering recommender systems.
- Seyerlehner, K., Knees, P., Schnitzer, D., and Widmer, G. (2009). Browsing music recommendation networks. In *ISMIR*, pages 129–134.
- Shani, G. and Gunawardana, A. (2009). Evaluating recommender systems. Technical Report MSR-TR-2009-159.
- Shardanand, U. (1994). Social information filtering for music recommendation.
- Shin, D., Lee, J.-w., Yeon, J., and Lee, S.-g. (2009). Context-aware recommendation by aggregating user context. In *Commerce and Enterprise Computing, 2009. CEC'09. IEEE Conference on*, pages 423–430. IEEE.

- Shyong, K., Frankowski, D., Riedl, J., *et al.* (2006). Do you trust your recommendations? an exploration of security and privacy issues in recommender systems. In *Emerging trends in information and communication security*, pages 14–29. Springer.
- Singhal, A. (2001). *Modern Information Retrieval: A Brief Overview*. Google. https://www.researchgate.net/publication/220282493_Modern_Information_Retrieval_A_Brief_Overview.
- Sommerville, I. (2010). *Software Engineering*. Addison-Wesley, Harlow, England, 9 edition.
- Symeonidis, P., Ruxanda, M. M., Nanopoulos, A., and Manolopoulos, Y. (2008). Ternary semantic analysis of social tags for personalized music recommendation. In *Ismir*, volume 8, pages 219–224.
- Tiemann, M. and Pauws, S. (2007). Towards ensemble learning for hybrid music recommendation. In *Proceedings of the 2007 ACM Conference on Recommender Systems, RecSys '07*, pages 177–178, New York, NY, USA. ACM.
- Wikipedia (2018). Phonograph record. Disponível em: https://en.wikipedia.org/wiki/Phonograph_record. Último acesso em 27 de Julho de 2018.
- Yoshii, K. and Goto, M. (2009). Continuous plsi and smoothing techniques for hybrid music recommendation. In *ISMIR*, pages 339–344. Citeseer.
- Yoshii, K., Goto, M., Komatani, K., Ogata, T., and Okuno, H. G. (2006). Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences. In *ISMIR*, volume 6, page 7th.