Algoritmo de mineração de dados e recomendação para mídias de entretenimento diversificadas

Eduardo da Silva Café [1]
Anderson Adaime de Borba [2]

[1]Faculdade de Computação e Informática (FCI)
Universidade Presbiteriana Mackenzie São Paulo, SP – Brasil

[2]Graduação em Sistemas de Informação
Faculdade de Computação e Informática (FCI)
Universidade Presbiteriana Mackenzie São Paulo, SP – Brasil

<10382738@mackenzista.com.br>
<anderson.borba@mackenzie.br>

10 de junho de 2024

Resumo

Esse trabalho buscou a criação de um algoritmo de recomendação para mídias de entretenimento em formatos diferentes como filmes, series, jogos e livros. Partindo da premissa de recomendar um jogo, por exemplo, baseado nos filmes e séries que uma pessoa consome. Para isso foi desenvolvido três etapas, sendo a primeira um algoritmo que acesse base de dados de serviços de cada mídia e traga esses dados a uma base de dados local centralizada facilitando o acesso. A segunda como o vínculo desses dados a pessoas onde o algoritmo será aplicado. E o terceiro a geração da recomendação baseada nos dados de consumo dessa pessoa utilizando o algoritmo Apriori onde foi analisado os valores e mostrado que é possível gerar recomendações.

Palavras-chave: Apriori;Recomendação; Big Data;

Abstract

This work seeks to create a recommendation algorithm for entertainment media in different formats, such as films, series, games, and books. They are starting from the premise of recommending a game, for example, based on the films and series that

a person consumes. To achieve this, three steps were developed, the first being an algorithm that accesses each media service database and brings this data to a centralized local database, facilitating access. The second is to link this data to people where the algorithm will be applied. The third is the generation of the recommendation based on that person's consumption data using the Apriori algorithm, where it was analyzed whether the values are valid for a good recommendation.

Keywords: Apriori; Recommendation; Big Data.

1 Introdução

1.1 Motivação

Com a popularização dos serviços vinculados a internet o acesso a mídias de entretenimento sejam series, filmes, livros, jogos etc. se tornaram mais acessíveis, nos dando muitas opções do que escolher para consumir, com esse avanço chegamos a uma nova problemática de termos muitas opções e nenhuma ideia de por onde começar a escolher, ainda mais se tratando dos diversos formatos de conteúdo voltado ao entretenimento.

Recentemente, vemos um crescimento nos mercados de serviços por *streamings* de filmes, series e jogos, assinaturas que dão acesso a diversos catálogos, base de livros que permitem leitura online de qualquer lugar. Uma facilidade de acesso a grandes conteúdos por preços acessíveis, e mais do que nunca inicia a incerteza de o que consumir e por onde começar, de todos os conteúdos disponíveis o que pode interessar ao consumidor.

Atualmente, boa parte das plataformas que atuam no ramo de mídias de entretenimento, sejam elas jogos, livros, filmes, series etc. possuem um algoritmo de recomendação para conteúdos presentes dentro de seus catálogos. Seja Netflix, Steam, Skoob, todos apresentam sistemas de recomendação de novos conteúdos. Mas quando se trata de sistemas de recomendação usando mais de uma dessas mídias temos uma falta de ferramentas conhecidas que consigam trabalhar com esse tipo de recomendação.

1.2 Objetivo

O objetivo desse trabalho foi criar um algoritmo de mineração de dados que acesse as bases individuais de diferentes mídias de entretenimento sejam elas jogos, filmes, séries ou livros e baseando-se no gosto pessoal de um indivíduo, consiga gerar uma recomendação de um segmento que ela não consome hoje.

Para realizar essa tarefa utilizamos o algoritmo Apriori de Mineração de Dados com a linguagem de programação Python, tendo como objetivo identificar padrões em diferentes bases de dados voltadas a mídias de entretenimento. Utilizar esses padrões como chave de associação entre as bases criando um relacionamento que permita recomendar um item de um outro segmento de entretenimento para uma pessoa.

Para acesso às bases de dados, usamos APIs *(application programming interface)* dos serviços provedores das mídias para levantamento geral da massa de dados, a guarda foi feita em uma base local de dados não relacional a fim de melhorar a eficiência no momento de realizar a consulta aos dados onde foi identificado os padrões através do algoritmo e por fim obtemos uma visão das relações para gerar a recomendação.

O resultado foi de conteúdo das 4 mídias citadas nesse trabalho (livros, filmes, series, jogos). Recebendo como entrada o que a pessoa consome hoje, estabelecendo um

perfil e por fim gerando a recomendação de quais conteúdos ela pode gostar mais nos diferentes segmentos.

2 Materiais e Métodos

2.1 Materiais

2.1.1 Linguagem de Programação

A escolha da Linguagem Python para desenvolver a aplicação de mineração de regras de associação ocorreu devido a ela ser uma das principais linguagens de programação voltada a Ciência de dados liderando o ranking de melhores linguagens para ciência de dados em 2022 segundo IEEE (CASS, 2022). Além de simplificar outras operações necessárias para esse trabalho como conectar a APIs externas para extração e armazenamento dos dados em uma base local, fazer uma limpeza de dados buscando o que é relevante para usarmos nesse algoritmo, além de ter boas ferramentas para representação dos resultados.

2.1.2 Controle de código GitHub

Como uma das principais ferramentas de controle de código atualmente a escolha do GitHub foi uma escolha certa permitindo o controle das versões e alterações do código, controle das etapas a serem desenvolvidas, documentação e disponibilidade do material desenvolvido.

A principal ferramenta disponibilizada do GitHub a ser utilizada é o repositório. (DOCS, 2024). Dentro do repositório foi guardado de todo código fonte do projeto que será desenvolvido, toda a documentação referente ao seu desenvolvimento e demais arquivos resultantes.

A organização do repositório seguiu o modelo desenvolvido por (FRERY, 2020).

Para gerenciamento das ações usamos a ferramenta do *GitHub Projects*. Essa ferramenta segue o formato de uma tabela onde podemos organizar cada etapa do projeto em campos personalizados podendo ser conectadas as solicitações de *push and pull* diretamente do código. (GITHUB, 2024)

Como uma ótima forma de visualização do andamento do projeto, essa ferramenta auxiliará no controle de cada etapa desse projeto, trazendo uma representação visual de cada etapa, seu status de desenvolvimento e documentação sobre o que esta sendo feito.

2.1.3 Base de dados

Segundo (CASTRO; FERRARI, 2016, p.25) "conceituamos base de dados como coleção organizada de dados, ou seja, valores quantitativos ou qualitativos referentes a um conjunto de itens, que permite uma recuperação eficiente dos dados. Conceitualmente, os dados podem ser entendidos como o nível mais básico de abstração a partir do qual a informação e, depois, os conhecimentos podem ser extraídos."

Dentro desse trabalho as bases de dados serão o objeto provedor de matéria prima para que possamos em cima delas trabalhar na geração de informação a partir dos dados que a compõem. Essa foi a primeira etapa desenvolvida nesse projeto.

2.1.3.1 Criação de uma base própria

Para facilitar o acesso aos dados uma técnica utilizada é salvar os resultados das requisições as APIs é armazena-los em uma base de dados local diminuindo o tempo de acesso e facilitando na criação de filtros para a etapa de mineração de dados. Podemos seguir dois modelos para criação de base de dados localmente sendo o Modelo Relacional e não Relacional.

O Modelo Relacional (DATE, 2004, p. 23) definiu que os dados seguem sempre a estrutura de tabelas onde qualquer obtenção de dados sempre retornará uma tabela que pode ser resultado da junção de várias outras tabelas. Esse modelo é muito usado para Bases de dados que trabalham com dados estruturados e padronizados facilitando encontrar relacionamentos.

O modelo não relacional (SILVA, 2021, p. 15) conforme citado por (SADALAGE; FOWLER, 2013) são bancos com estrutura de dados livres oferecendo maior flexibilidade em relação a forma como os dados são salvos. Esse modelo é usado principalmente para dados semi estruturados e não estruturados quando não se tem conhecimento sobre sua composição.

Esse projeto para as mídias de entretenimento utiliza dados vindo de API com estruturas e informações diferentes para cada mídia. A base de dados de jogos terá informações diferentes da base de livros assim tornando mais simples utilizar o modelo não relacional para armazenamento desses dados.

2.1.3.2 Banco de dados MongoDb

Como lidamos com grandes massas de dados vindas de diversas bases diferentes, com tipos e formatos diferentes, para melhoria na eficiência a consultar essas massas usaremos o serviço de bancos de dados não relacional mongoDb.

A vantagem desse modelo de banco de dados como trabalhamos com dados de tipos diferentes e não temos visão a princípio do relacionamento o armazenamento por uma base não relacional acaba sendo mais interessante a princípio nessa aplicação, como não temos uma padronização nos tipos de dados que recebemos.

A escolha do banco de dados MongoDb como base de dados não relacional escolhida para esse projeto é devido a ser a base mais popular seguindo o modelo não relacional, tendo grande flexibilidade quanto ao modelo de dados, facilidade na configuração e tendo um bom desempenho.

2.1.4 Acesso

Como primeiro método de acesso aos dados utilizamos APIs de serviços que atuam com mídias de entretenimento. Conceitua-se API como uma interface de programação que permite outros aplicativos acessarem uma determinada aplicação seguindo as regras dessa aplicação. (FERREIRA, 2021, p. 14).

A maior parte dos serviços online disponibiliza suas APIs para acesso sem custos, sendo apenas necessária a criação de uma chave de autenticação dentro da plataforma para ter acesso pela aplicação. E com isso obteve-se acesso a dados dentro da plataforma que foram utilizados para esse trabalho.

Com isso foi realizado a extração das massas de dados necessário de cada mídia por meio desse método de comunicação, apresentando apenas limitações de tempo de requisição e restrições de filtros de dados a serem usados. Com isso entramos na próxima etapa da base de dados própria para armazenamento das requisições de cada API.

A leitura de arquivos acaba por ser mais simples pois os dados estão salvos localmente não demandando conexão com a internet para ter acesso, uma vez que estejam baixados, podendo assim fazermos a leitura uma única vez, sendo também sua desvantagem já que a atualização de novos itens tem que ser feita manualmente baixando novos arquivos e carregando seus dados na base do projeto.

2.1.4.1 Steam API

A Steam sendo um dos principais nomes quando falamos de mercado de jogos. nos traz muitos recursos que auxiliam seus clientes desde informações detalhadas de jogos, dados de consumo e uma grande gama de usuários. Segundo seu acordo de licença As APIs Web da Steam são serviços que permitem a pessoas licenciadas obterem determinados dados da plataforma Steam tendo uma Chave de Api. (VALVE, 2010)

A Api nos fornece métodos de listagens dos jogos presentes na plataforma nos dando um catálogo extenso para trabalharmos, métodos para trazer dados detalhados dos jogos onde buscamos uma visão de relacionamento para poder gerar a recomendação e pôr fim a opção de trazer dados de consumo de um usuário.

2.1.4.2 The Movie Database API

The Movie Database é uma base popular de filmes e séries gratuita desenvolvida em código aberto que possuí um catálogo extensivo sobre dados de filmes e séries. Sendo aplicada em diversos mercados internacionais trazendo confiabilidade nas informações que disponibiliza. Serviço disponibiliza abertamente dados de filmes e séries para que outras aplicações possam realizar buscas de dados ou imagens. (THEMOVIEDB, 2023)

Ela traz detalhes sobre filmes e séries o que facilitará o algoritmo a encontrar regras de associações entre as mídias tendo mais dados detalhados para trabalhar. Além de diversos métodos para ordenar a busca como filmes e séries lançados recentemente, mais famosos, semelhantes a um título e outros diversos. O The Movie Database disponibiliza uma lista com os ids para baixarmos não sendo necessário buscar diretamente na base ao contrário da Steam. Com isso a leitura dos ids foi feita através do arquivo local, e para buscar os dados específicos de cada filme e serie individualmente dentro da Api.

2.1.4.3 Open Library

Assim como o The Movie Database a Open Library se trata de um acervo digital que tem como principal objetivo de disponibilizar todas as obras existentes no mundo para acesso facilitado. A Open Library oferece dois tipos de serviços sendo uma API para obtenção de dados em baixa frequência e despejos de arquivos mensais para projetos que demandem consumo maior de dados. (OPENLIBRARY, 2021)

No caso da Open Library a utilização foi em maior parte dos despejos de dados onde obtivemos acesso aos dados dos livros presentes em seu catálogo, com o uso da API sendo voltado apenas para recuperação de alguns dados. Por se tratar de uma leitura de

arquivos foi necessário a entrada periodicamente para trazer atualizações a essa massa de dados referente aos livros, uma vez que sua API não é feita para grandes extrações. Com isso foi necessário baixar os arquivos mais recentes disponibilizados com dados atualizados dos livros, edições e autores.

Por não termos integração com Api. Apenas baixamos o arquivo que possuí a autores, edições de livros, revisões e autores.

Fizemos uma limpeza inicial para remover os autores, edições e revisões devido a eles não serem utilizados nesse trabalho e salvamos o restante na nossa base de dados para facilitar o acesso.

Esse serviço a princípio foi alterado a forma como foi usado por não trazer metadados que permitissem o vínculo com filmes, séries e jogos, então estamos usando-o para trazer os dados de ISBN para buscar no google API.

2.1.4.4 ISBN

Segundo a Camará Brasileira do Livro "O ISBN (International Standard Book Number/ Padrão Internacional de Numeração de Livro) é um padrão numérico criado com o objetivo de fornecer uma espécie de "RG"para publicações monográficas, como livros, artigos e apostilas. A difusão global do ISBN e a facilidade com que é lido por redes de varejo, bibliotecas e sistemas gerais de catalogação, tornou-o imprescindível para qualquer publicação."(LIVRO, 2023)

Usamos o ISBN para realizar a ponte entre a lista de dados da Open Library com o google API. Pois como se trata de um identificador global para cada livro no mundo, podemos utilizá-lo como código único para eles e fazer a ponte entre os serviços mesmo que cada um tenha informações diferentes sobre livros.

2.1.4.5 Google Books API

Assim como nas outras APIs foi necessário criar uma chave de API para podermos ter acesso ao serviço. A plataforma do Google mostrou um pouco de dificuldade para descobrir como conseguir informações sobre manuseio de sua API.

O primeiro passo foi acessar a página <https://console.cloud.google.com/> e criar um projeto de desenvolvimento. Segundo passo foi escolher o serviço que queria usar no caso o Google Books e ativar o serviço dentro do meu projeto com isso gerando uma chave de API para acessá-lo pela minha aplicação.

Com a chave de API foi usado o método <https://www.googleapis.com/books/v1/volumes> onde conseguimos buscar por volumes de livros específicos na API. Esse *endpoint* pede 3 parâmetros a serem preenchidos. O parâmetro "q"onde você preenche uma informação que deseja buscar na API, nesse caso usamos o número de ISBN do livro obtido pela Open Library. O parâmetro "maxResults"que você informa a quantidade máxima de itens a serem retornados aonde no meu caso passei apenas 1, podendo usar no máximo 40 resultados. E o parâmetro "key"onde você passa a sua chave de API criada anteriormente.

A API do google não disponibiliza um método que conseguimos listas de livros sem nenhuma informação por isso tivemos que voltar nos livros obtidos pela Open Library pois nela tínhamos uma lista de livros com o atributo ISBN que é um identificador global

para livros e com isso usamos o ISBN para buscar por cada livro específico na API do Google Livros e trazer os metadados que precisávamos para o trabalho.

2.2 Métodos

2.2.1 Mineração de dados

Para (PIZZI, 2006, p. 15) "Mineração de dados consiste na extração de conhecimento a partir de grandes quantidades de dados". esse processo consiste em esforços para descobrir padrões dentro de bases de dados gerando conhecimento útil para tomada de decisão (SILVA; PERES, 2015, p. 08).

A mineração de dados é o procedimento principal usado dentro desse trabalho, pois o Apriori é uma dessas técnicas de mineração de dados que nesse caso é destinado a geração de uma recomendação.

2.2.2 Processo de Descoberta de conhecimento em bases

O processo de mineração de dados é um dos integrantes do processo de descoberta de conhecimento em bases de dados do inglês *"Knowledge Discovery in Databases- KDD"* Para Silva (2016) O KDD se trata de processos analíticos, sistemáticos e se possível automatizados voltados a descoberta de conhecimento dentro das bases. Segundo (SILVA; PERES, 2015, p. 07). O KDD é composto pelos procedimentos de escolha da base de dados para estudo, limpeza inicial da base, seleção dos dados para a problemática desejada e tratamento para o problema. e por fim mineração e avaliação dos resultados. (CASTRO; FERRARI, 2016, p. 26 - p.27)

2.2.2.1 Seleção e integração das bases de dados

Sendo a primeira etapa do processo de KDD. Essa etapa é definida como a delimitação de quais dados serão usados para obter o conhecimento necessário. (SILVA; PERES, 2015, p. 07)

Essa etapa como destacado anteriormente foi encontrar os provedores de dados, atendimento de todos os requisitos para ter acesso e identificação dos parâmetros e filtros necessários a fim de obter um bom montante de dados para formar a base local onde são feitas as consultas para o algoritmo de mineração trabalhar.

2.2.2.2 Pré processamento

Feito após a consulta das bases externa. O pré processamento é onde prepara inicialmente os dados para análise. (CASTRO; FERRARI, 2016, p. 48). "Problemas típicos associados à baixa qualidade de dados são ausência de valores, dados ruidosos, valores inconsistentes, atributos de naturezas distintas e redundância de dados. Muitas dessas características não são evidentes no conjunto de dados, mas podem ser reveladas por procedimentos da análise exploratória". (SILVA; PERES, 2015, p. 47)

A execução de um algoritmo de pré-processamento nesse trabalho foi usado para preparar os dados vindos do banco de dados não relacional e adaptá-los a forma como o Apriori trabalha.

2.2.2.3 Avaliação dos dados

Como último procedimento do KDD a avaliação de dados é onde de acordo com o objetivo da análise confirma se o conhecimento retirado dos dados é relevante a esse contexto.(CASTRO; FERRARI, 2016, p. 27)

Essa etapa varia de acordo com o objetivo do algoritmo, tendo suas métricas de avaliação sendo definidas pelos criadores do algoritmo e pela regra de negócio no qual ele está inserido.

2.2.3 Caracterização de informações obtidas

Os algoritmos de mineração de dados trabalham em buscar por duas características dentro das massas. Podendo ser classificadas como descritivas e preditivas. Cada uma gerando um conhecimento diferente das massas.

Na análise descritiva. Para (SILVA; PERES, 2015, p. 27) ela nos mostra as principais características presentes naquele conjunto de dados podendo o descrever ou resumir suas características.

Nessa análise, o algoritmo realiza a busca das relações entre os diferentes tipos de dados, onde nela obteremos os padrões de vínculo entre as bases diferentes conseguindo as regras de associação.

Enquanto na análise preditiva. Para (CASTRO; FERRARI, 2016, p. 29) ela é voltada a classificação dos dados ou no processo de estimar possíveis valores que possam compor esse conjunto de dados.

Na análise preditiva nós verificamos se os padrões encontrados dentro da análise descritiva são válidos e estabelecem uma conexão entre as bases e através desses padrões podemos estabelecer a conexão entre as bases das mídias e gerarmos as recomendações.

2.2.4 Algoritmo Apriori

Um dos algoritmos mais comuns e eficientes na mineração de dados e recomendação é o Algoritmo Apriori. Para (PIZZI, 2006, p. 18 - p.19), conforme citado por (AGRAWAL; SRIKANT, 1994) “é um algoritmo que minera regras de associação booleanas. É um algoritmo iterativo que busca por k-itemsets freqüentes a partir dos (k-1)-itemsets. Para isso, é usada a "propriedade Apriori", que se baseia no fato de que um itemset freqüente não pode conter um subconjunto não frequente”.

Sendo um dos pioneiros e principais algoritmos de mineração de regras de associação, ele nos traz uma confiabilidade para aplicarmos na proposta desse projeto para minerar as regras de associação dentro das bases sendo que grande parte dos algoritmos subsequentes o utilizam como base.

2.2.4.1 Itens frequentes

O algoritmo Apriori parte de uma lógica simples de desenvolvimento. Segundo (SILVA; PERES, 2015, p. 206). “Esse algoritmo é composto por duas fases. Na primeira (Algoritmo 5-1), é verificada a quantidade de vezes que cada item (ou 1-itemset) ocorre na base de dados transacional (TID), ou seja, o suporte de cada item é calculado. Os itens com suporte maior que o mínimo estabelecido pelo usuário são selecionados e combinados,

de forma a compor 2-itemsets, e o suporte desses novos itemsets é calculado. Aqueles de suporte maior ou igual ao suporte mínimo são então selecionados. Na sequência, os itemsets selecionados (2-itemsets) são novamente combinados, formando os 3-itemsets, e todo o processo é repetido até que, em uma iteração, não seja produzido nenhum itemset frequente. A cada iteração (k) desse processo, uma lista de itemsets frequentes (L_k) é produzida. Ao final, as listas (L_k) são concatenadas na lista L de itemsets frequentes.".

Ao final buscamos entender quais itens são frequentes nos conjuntos de dados analisados, assim estabelecendo uma relação de quem consome uma coisa tem chance de consumir outra junto. Por exemplo quem consome o livro "Harry Potter"tem chance de gostar do jogo "Príncipe da Persia", porque o algoritmo encontrou relação entre os dois produtos.

2.2.4.2 Regras de associação

Segundo (CASTRO; FERRARI, 2016, p. 270). "As regras de associação são obtidas após a determinação dos conjuntos de itens frequentes F_k. Cada conjunto frequente F_k pode gerar até $2^k - 2$ regras de associação, desconsiderando aquelas com antecedente ou consequente nulos. Uma regra de associação pode ser obtida particionando F_k em dois conjuntos não vazios, A e $(B - A)$, tal que $A \rightarrow (B - A)$ satisfaça minconf. Como essas regras fazem parte do conjunto de itens frequentes, elas já satisfizeram o limiar de suporte, minsup. Para cada item frequente $Y = \{i_1, i_2, \cdots, i_k\} \in F, k \geq 2$, pode-se gerar todas as regras (no máximo k) que usam itens do conjunto Y. O antecedente de cada regra será o subconjunto A de C tal que $|A| < k$ e o consequente será o item $C - A$.".

As regras de associação são o objetivo desse trabalho, onde identificamos os itens frequentes dentro do conjunto de dados, podendo estabelecer as regras que criam relações entre esses dados e avaliamos a força dessas regras para gerar a recomendação.

2.2.4.3 Métricas para o algoritmo.

A primeira métrica que utiliza no Apriori para avaliar os resultados é o suporte. Essa métrica é usada para saber a porcentagem de quantas vezes um item, ou conjunto de itens apareceram no *Dataset* . Sendo a principal informação para determinar as métricas (CASTRO; FERRARI, 2016, p. 260). Fórmula para o suporte.

$$(A -> C) = \frac{(A \rightarrow C)}{n}$$

A confiança é uma métrica que define o quanto o item consequente apareceu nas ocorrências do item antecedente determinando o quanto eles estão unidos criando essa regra de associação. (CASTRO; FERRARI, 2016, p. 260). A fórmula para calcular a confiança é a seguinte.

$$(A -> C) = \frac{A \rightarrow C}{A}$$

O Lift é uma medida calculada a partir da confiança e suporte afim de determinar se a relação entre os itens antecedentes e consequentes aumentam a possibilidade de saída um do outro. (CASTRO; FERRARI, 2016, p. 261) O lift é uma medida importante pois calcula a chance de dois se alavancarem entre si. Um Lift maior que 1 indica que se um dos

itens foi consumido o segundo tem uma boa chance de também ser consumido. A fórmula para calcular o Lift é.

$$(A- > C) = \frac{conf(A \rightarrow C)}{sup(A) \times sup(C)}$$

A convicção já se calcula a partir do suporte do item antecedente junto a confiança do consequente onde se identifica se os itens são dependentes entre si. (CASTRO; FERRARI, 2016, p. 261). A convicção é outra medida boa que informa o quanto um item é dependente do outro para a recomendação. Quanto maior a convicção maior a dependência entre esses itens tendo como o ideal ela ser maior que 1. A fórmula para calcular a convicção é.

$$(A- > C) = \frac{1 - sup(C)}{1 - sup(A \rightarrow C)}$$

2.2.4.4 Aplicação

Para execução desse algoritmo ele recebe como entrada os dados uma tabela composta por cada linha sendo um usuário e cada coluna um item de uma das 4 mídias analisadas nesse trabalho onde o algoritmo busca os relacionamentos nos itens de acordo com o que cada pessoa consumiu. Por fim retornando os relacionamentos entre cada item permitindo assim gerar a recomendação

Uma das principais aplicações do Apriori em serviços é na recomendação de item que várias pessoas consumiram em comum. Onde por exemplo em um serviço de *streaming* que 10 pessoas assistiram ao filme x também assistiram a série y, a pessoa tendo assistido o filme x tem uma boa probabilidade de gostar da série y.

Essa seria a forma de aplicar o algoritmo dentro desse trabalho, juntando todos os filmes, livros, series e jogos que uma pessoa x consumiu, e baseado no que outras pessoas consumiram em comum gerar a recomendação sobre quais itens mais aparecem juntos.

Abaixo na tabela 1 é apresentado um exemplo criado pelo autor sobre como seria essa estrutura baseando-se no que usuários consumiram ou não sendo "verdadeiro"como consumiu e "falso"como não consumiu. Tendo cada coluna como uma mídia de entretenimento e cada linha como um usuário.

Consumiu?	filme 1	serie 1	jogo 1	livro 1	filme 2
usuário 1	verdadeiro	verdadeiro	falso	verdadeiro	falso
usuário 2	verdadeiro	falso	falso	verdadeiro	falso
usuário 3	falso	falso	verdadeiro	falso	verdadeiro
usuário 4	verdadeiro	verdadeiro	falso	falso	falso

Tabela 1 – Modelo dos dados a serem enviados ao Apriori

O modelo irá verificar quais de cada mídia mais aparecem para os usuários onde pegamos o exemplo do usuário que consumiu o filme 1, a série 1 e o livro 1. E queremos fazer uma recomendação para o usuário 4 que apenas consumiu o filme 1 e a série 1. Levando em consideração que tanto o usuário 1 quando o usuário 2, consumiram o filme 1 e o livro 1. Podemos estabelecer que para o usuário 4, o mais recomendado seria o livro 1, pois o usuário 1 e 2 que consumiram o mesmo filme que ele também consumiram aquele livro. Ao contrário do jogo 1 e filme que foi consumido apenas pelo usuário 3. Que não consumiu nem o filme 1 nem a série 1 que o usuário 4.

Abaixo na tabela 2 é um exemplo de como deve ser a saída do Apriori para esse modelo por pessoas.

Mídia	Suporte
filme 1	75,00
serie 1	50,00
livro 1	50,00
jogo 1	25,00
filme 2	25,00

Tabela 2 – Modelo do Resultado do Apriori baseado na tabela 1.

3 Construção do algoritmo

Os resultados desse projeto estão divididos em três etapas. A criação da base de dados com as mídias, a criação dos usuários e a geração das recomendações

3.1 Criação de uma base de dados com todas as mídias

A primeira etapa desse projeto se resume a obtenção dos dados de jogos, filmes, livros e séries. Para isso utilizamos os serviços de APIs citados anteriormente para obter esses dados e salvá-los em uma base local para facilitar seu acesso.

Para armazenamento foi criado uma coleção separada no mongodb para filmes, uma para series, uma para jogos e uma para livros facilitando saber a origem de cada informação e como resultado obtivemos um conjunto de 20367 filmes, 12987 jogos, 13039 livros e 20133 series que foram usados como amostras de cada usuário para gerar a recomendação.

3.2 Criação de usuários

Após termos a base de dados precisamos criar os usuários e vincular eles a itens dentro das bases de mídias de entretenimento para gerar os perfis de consumo que o Apriori usará para gerar a recomendação.

Para isso criamos uma função que obterá de 0 a 250 itens dentro de cada base de filmes, livros, series e jogos. (Por exemplo de 80 filmes, 50 jogos, 13 livros e 143 series) e salvara os ids de cada item em uma lista vinculada a um usuário da base e assim popularemos uma coleção de usuários com esse formato onde o Apriori atuará.

O resultado desse algoritmo foi uma base de dados com 5368408 usuários cada um com uma lista que pode variar de 0 a 250 itens de filmes, livros, series e jogos que serão usados para rodar o Apriori.

Foi Necessário a criação dessa base de dados devido a esse trabalho não ter acesso nem expor dados pessoais de pessoas reais. Assim criando uma máscara para os dados reais.

3.3 Gerando a recomendação

Nessa última etapa aplicaremos o algoritmo apriori para fazer as recomendações nessa base de dados. O Algoritmo criado para fazer isso foi divido nas seguintes etapas.

Primeira etapa foi a obtenção dos dados na base de dados para o algoritmo. Depois foi feita a junção dos dados das listas de filmes, livros, jogos e series em uma única lista chamada de mídias que será passada para o algoritmo.

A próxima etapa foi converter os dados do mongoDb para rodar no Apriori onde o transformamos em um Dataframe do Python constituído por cada coluna o id de uma mídia e cada linha um usuário diferente e o seu preenchimento sendo Como Verdadeiro ou Falso para se ele consumiu ou não.

A 3ª etapa onde rodamos o Apriori para esse Dataframe e obtemos os resultados do algoritmo para quais itens são mais frequentes e quais aparecem em conjunto com mais frequência que outros itens.

E por fim a partir dos resultados do Apriori calculamos as outras medidas do algoritmo que são a Confiança, o Lift e a Convicção e com bases nesses resultados junto ao suporte já retornado anteriormente pelo Apriori avaliamos se os valores geram uma boa recomendação.

4 Resultados

4.1 Base Simulada

Para validação dos resultados do algoritmo foi feito uma coleção no mongo DB com 11 usuários colocados manualmente para simulação dos resultados. Que gerou o seguinte Dataframe mostrado na tabela 3.

	Doctor Who	Merlin	GTA V	MobyDick	Sherlock	Os Familiares	Corra	A esperança	Hereditário	Extraordinário
Usuário 1	Verdadeiro	Falso	Falso	Verdadeiro	Falso	Falso	Verdadeiro	Verdadeiro	Falso	Falso
Usuário 2	Falso	Falso	Falso	Verdadeiro	Falso	Verdadeiro	Verdadeiro	Verdadeiro	Verdadeiro	Falso
Usuário 3	Falso	Verdadeiro	Verdadeiro	Verdadeiro	Falso	Falso	Verdadeiro	Verdadeiro	Falso	Verdadeiro
Usuário 4	Falso	Verdadeiro	Verdadeiro	Verdadeiro	Verdadeiro	Falso	Falso	Falso	Verdadeiro	Verdadeiro
Usuário 5	Falso	Verdadeiro	Verdadeiro	Verdadeiro	Falso	Falso	Verdadeiro	Falso	Falso	Verdadeiro
Usuário 6	Falso	Falso	Verdadeiro	Falso	Falso	Verdadeiro	Falso	Verdadeiro	Verdadeiro	Falso
Usuário 7	Verdadeiro	Falso	Verdadeiro	Verdadeiro	Falso	Verdadeiro	Verdadeiro	Verdadeiro	Falso	Verdadeiro
Usuário 8	Verdadeiro	Verdadeiro	Falso	Falso	Verdadeiro	Verdadeiro	Falso	Falso	Verdadeiro	Verdadeiro
Usuário 9	Falso	Verdadeiro	Verdadeiro	Verdadeiro	Falso	Verdadeiro	Falso	Falso	Falso	Verdadeiro
Usuário 10	Verdadeiro	Falso	Verdadeiro	Verdadeiro	Falso	Verdadeiro	Verdadeiro	Verdadeiro	Verdadeiro	Falso
Usuário 11	Falso	Falso	Verdadeiro	Falso	Falso	Verdadeiro	Falso	Verdadeiro	Falso	Falso

Tabela 3 – Dataframe base simulada.

Como resultado obtivemos a seguinte saída para esse Dataframe 4 simulado, ajustando o suporte mínimo do algoritmo para 0,4.

Antecedentes	Consequentes	Suporte Antecedente	Suporte Consequente	Suporte	Confiança	Lift	Conviction
Extraordinário	merlin	0.54545	0.45455	0.45455	0.83333	1.83333	3.27273
merlin	Extraordinário	0.45455	0.54545	0.45455	1.0	1.83333	inf
Mobydick	GTA V	0.72727	0.72727	0.54545	0.75	1.03125	1.09091
GTA V	MobYdick	0.72727	0.72727	0.54545	0.75	1.03125	1.09091
Os Familiares	GTA V	0.63636	0.72727	0.45455	0.71429	0.98214	0.95455
GTA V	Os Familiares	0.72727	0.63636	0.45455	0.625	0.98214	0.9697
A esperança	GTA V	0.63636	0.72727	0.45455	0.71429	0.98214	0.95455
GTA V	A esperança	0.72727	0.63636	0.45455	0.625	0.98214	0.9697
Extraordinário	GTA V	0.54545	0.72727	0.45455	0.83333	1.14583	1.63636
GTA V	Extraordinário	0.72727	0.54545	0.45455	0.625	1.14583	1.21212
Mobydick	Corra	0.72727	0.54545	0.54545	0.75	1.375	1.81818
Corra	Mobydick	0.54545	0.72727	0.54545	1.0	1.375	inf
A esperança	Mobydick	0.63636	0.72727	0.45455	0.71429	0.98214	0.95455
mobidick	A esperança	0.72727	0.63636	0.45455	0.625	0.98214	0.9697
Extraordinário	Mobydick	0.54545	0.72727	0.45455	0.83333	1.14583	1.63636
Mobydick	Extraordinário	0.72727	0.54545	0.45455	0.625	1.14583	1.21212
A esperança	Os Familiares	0.63636	0.63636	0.45455	0.71429	1.12245	1.27273
Os Familiares	A esperança	0.63636	0.63636	0.45455	0.71429	1.12245	1.27273
A esperança	Corra	0.63636	0.54545	0.45455	0.71429	1.30952	1.59091
Corra	A esperança	0.54545	0.63636	0.45455	0.83333	1.30952	2.18182

Tabela 4 – Resultados da associação de itens.

Com esses resultados podemos utilizá-los para gerar as seguintes recomendações. "GTA V"e "Mobidick"apareceram 8 vezes na análise desses 11 usuários calculando o suporte obtemos a porcentagem de 8/11 = 72,72% de aparições fazendo com que sejam ótimas opções para recomendação. A confiança entre esses dois itens foi de 75% em ambos os casos devido as 8 pessoas que consumiram "GTA V"e "Mobidick"6 consumiram os dois juntos logo 6/8 = 0,75 ou 75% O Lift foi de 1,03125 devido a 0,54 / (0,72 * 0,72) = 1,03125 indicando uma alavancagem dos dois itens devido ao Lift ser maior que 1. E a convicção dos dois itens foi de 1,09091 devido a (1 - 0,72727) / (1 - 0,75) = 1,09091 indicando que existe uma dependência dos dois itens por ela estar acima de 1, porém essa dependência sendo baixa.

Obtivemos também "Corra"e "Mobydick"com "Mobydick"aparecendo com 8 pessoas e "Corra"aparecendo para 6 pessoas e ambos aparecendo juntos em 6 dos 11 usuários com porcentagem de 54,54% dos resultados. A confiança foi de 0,75% de "Mobydick"como antecedente com "Corra"como consequente devido a 6/8 = 0,75 ou 75% e 100% de "Corra"como antecedente para "Modybick"como consequente devido a 6/6 = 1 ou 100% resultando em todas as pessoas que assistiram "Corra"leram "Mobydick". O Lift foi de 1,375 devido a 0,54545 / (0,72727 * 0,54545) = 1,375 resultando em uma boa alavancagem dos dois itens. E por fim a convicção que de "Mobydick"para "Corra"como antecedente foi 1,81818 devido a (1-0,54545) / (1-0,75) = 1,81818 e de "Corra"para "Mobydick"como consequente ser infinito devido a (1-0,72727) / (1-1) = infinito, indicando que Corra é totalmente dependente de "Mobydick"por estar em todas as transações de "Mobydick"e "Mobydick"tem alta dependência de "Corra". Isso faz com que chegamos a quem costuma ler "Mobydick"também assistiu "Corra".

Assim com esses resultados podemos confirmar o funcionamento desse algoritmo indicando que a maioria dos resultados são considerados validos para gerar uma boa recomendação de conteúdo.

4.2 Base de dados Real

Com o algoritmo validado resta apenas aplicá-lo na base de dados real e verificar seus resultados. Para essa base foi necessário reduzir o suporte mínimo do algoritmo para 0,001 devido a ter uma variação maior de mídias entre cada usuário.

4.2.1 Resultados com os 10000 primeiros usuários.

Primeiramente o algoritmo foi aplicado para os primeiros 10000 itens da coleção usuários com o suporte mínimo em 0.001. Obtivemos uma tabela com 4328525 resultados e então foram retirado alguns itens para análise dos resultados do algoritmo mostrados na tabela 5.

Antecedentes	Consequentes	Suporte Antecedente	Suporte Consequente	Suporte	Confiança	Lift	Conviction
Play School serie	Doctor Who serie	0.2052	0.0676	0.0174	0.0848	1.2544	1.0188
Doctor Who serie	Play School serie	0.0676	0.2052	0.0174	0.2574	1.2544	1.07035
Play School serie	Flying Aces - Navy Pilot Simulator jogo	0.2052	0.0237	0.0058	0.02827	1.19262	1.00470
Flying Aces - Navy Pilot Simulator jogo	Play School serie	0.0237	0.2052	0.0058	0.24473	1.19262	1.05233
Wing Commander filme	Essays livro	0.0091	0.0548	0.0014	0.15385	2.80741	1.11705
Essays livro	Wing Commander filme	0.0548	0.0091	0.0014	0.025547	2.80741	1.01688

Tabela 5 – Resultados para os 10000 primeiros usuários.

Analisando os resultados obtivemos a série "Play School"como a que mais aparece nesses 10000 primeiros itens aparecendo em 20,52% dos resultados. Para esse modelo de dados resultou na série sendo bem recomendada juntamente a outros itens. Como no exemplo em que "Doctor Who"a segunda série que mais apareceu com 6,76% dessas pessoas. E para esse caso os dois itens aparecem juntos em 1,74% das transações, outra métrica que podemos analisar é a confiança entre esses dois itens sendo que 8,48% das pessoas que assistiram "Play School"também assistiram "Doctor Who"e 25,74% das pessoas que assistiram "Doctor Who"assistiram "Play School". Outro valor para analisarmos é o Lift ou alavancagem que nesse caso é de 1,2544 o que faz com que um desses itens aumente a chance de gostar do outro. A Convicção de "Play School"como antecedente para "Doctor Who"como consequente foi de 1,0188 enquanto de "Doctor who"como antecedente para "Play School"como consequente foi de 1,0703 o que indica que em ambos os casos se uma pessoa gostou de uma dessas séries existe uma boa probabilidade de ela gostar da outra.

Verificando ao primeiro item dos resultados de uma mídia diferente temos o relacionamento entre o jogo "Flying Aces - Navy Pilot "e novamente a serie "Play School". Onde o jogo "Flying Aces - Navy Pilot Simulator"aparece em 2,37% dos 10000 usuários analisados os dois aparecem juntos em 0,58% dos resultados, a confiança entre "Flying Aces - Navy Pilot Simulator"como antecedente para "Play School"como consequente é de 24,47% enquanto "Play School"como antecedente para "Flying Aces - Navy Pilot Simulator"como consequente é de 8,28%. O lift desses dois itens é de 1,1926 o que mostra ser um bom resultado devido ao lift ser maior que 1. E a convicção de "Flying Aces - Navy Pilot Simulator"como antecedente para "Play School"como consequente é de 1,0523 enquanto de "Play School"como antecedenet para "Flying Aces - Navy Pilot Simulator"como consequente é de 1,0174 também mostrando um bom resultado sendo maior que 1. Mostrando com exceção do suporte valores próximos a comparação entre "Play School"e "Doctor Who"mostrando uma boa recomendação.

Analisando ao primeiro livro e filme chegamos ao livro "Essays"e o Filme "Wing Commander"tendo dos 10000 usuários analisados houve 91 pessoas que assistiram ao filme "Wing Commander"e 548 pessoas que leram o livro "Essays"e desse grupo 14 pessoas consumiram os dois juntos. A confiança de "Wing Commander"como antecedente para "Essays"como consequente foi de 15,39% e de "Essays"como antecedente para "Wing Commander"como consequente foi de 2,55%. O lift desses dois itens foi de 2.8074 e a convicção de "Wing Commander"como antecedente para o "Essays"como consequente foi de 1,1171 enquanto de "Essays"como antecedente para "Wing Commander"como consequente foi de 1,0169. Nesse caso obtivemos um suporte e uma confiança menor que nos exemplos anteriores, mas devido a terem aparecido em poucas transações o lift foi maior que

nos anteriores e a convicção ainda foi maior do que 1, ainda resultando em uma boa recomendação.

4.2.2 Resultados para os 50000 primeiros usuários

Depois rodamos para os 50000 primeiros usuários com o suporte mínimo em 0,001 como aumentaram o número de dados analisados obtivemos menos resultados que para 10000, pois o algoritmo no modelo de 10000 considerou 10 relacionamentos mínimos para dois itens como resultado enquanto esse considerou 50 relacionamentos mínimos, tendo 1677152 resultados para essa execução, assim como no Anterior foi separado alguns desses resultados para análise mostrados na tabela 6.

Antecedents	Consequents	Antecedent Support	Consequent Support	Support	Confidence	Lift	Conviction
Play School serie	The Beverly Hillbillies serie	0.20198	0.0034	0.00108	0.00535	1.57267	1.00196
The Beverly Hillbillies serie	Play School serie	0.0034	0.20198	0.00108	0.31765	1.57267	1.16951
Play School serie	The Fly II filme	0.20198	0.00762	0.00168	0.00832	1.09156	1.00070
The Fly II filme	Play School serie	0.00762	0.20198	0.00168	0.22047	1.09156	1.02372
Homam filme	Democracy 3 jogo	0.01308	0.05448	0.001	0.07645	1.40331	1.02379
Democracy 3 jogo	Homam filme	0.05448	0.01308	0.001	0.01836	1.4033	1.00537
Lectures, addresses livro	Monsters vs Aliens filme	0.05142	0.01594	0.00106	0.02061	1.29326	1.00477
Monsters vs Aliens filme	Lectures, addresses livro	0.01594	0.05142	0.00106	0.06650	1.29326	1.01615

Tabela 6 – Resultados para 50000

Chegando para os 50000 os primeiros resultados foi uma associação entre a serie "Play School"e a serie "The Beverly Hillbillies". Assim como no caso anterior "Play School"foi o item que mais apareceu dentro do *dataset* o colocando na maioria das recomendações e comparando com"The Beverly Hillbillies". Obtemos um suporte de 0,11%, ou seja, de 50000 pessoas analisadas 54 consumiram os dois itens juntos. A confiança de "Play School"para "The Beverly Hillbillies"foi de 0,54% enquanto de "The Beverly Hillbillies"como antecedente para "Play School"como consequente foi de 31,77% Esses resultados são devido a "Play School"estar em grande presença no *dataset* com 20,20% dos 50000 usuários terem assistido "Play School"e apenas 0,34% dos 50000 terem assistido "The Beverly Hillbillies"resultando em uma confiança maior de "The Beverly Hillbillies"para "Play School". O lift para esses dois itens foi de 1,57267 resultando nos dois itens se alavancarem entre si. Enquanto analisando a Convicção vemos que de "Play School"como antecedente para "The Beverly Hillbillies"como consequente temos 1,00196 e de "The Beverly Hillbillies"como antecedente para "Play School"como consequente foi de 1,16951 revelando que "The Beverly Hillbillies"tem uma dependência com "Play School", porém "Play School"não tem uma boa dependência com "The Beverly Hillbillies"devido a mesma questão citada anteriormente, mas como os valores ainda estão acima de 1 ainda existe uma dependência.

Obtendo os primeiros resultados de itens de mídias diferente chegamos a um filme e uma série nesse caso o filme "The Fly II"e a série "Play School". O suporte dos dois itens juntos foi de 0,17%, ou seja, dos 50000 usuários analisados 84 pessoas consumiram esses dois itens juntos. A confiança de "Play School"como antecedente para "The Fly II"como consequente foi de 0,83% enquanto de "The Fly II"como antecedente para "Play School"como consequente foi de 22,05% Esses resultados são esperados devido a igual os resultados anteriores "Play School"estar presente para muitos usuários nesses resultados causando uma confiança alta de "The Fly II"para "Play School", porém baixa de "Play School"para "The Fly II". O Lift desses dois foi de 1,09156 tendo uma alavancagem menor que o item anterior, mas ainda sendo acima de 1 ou seja um item alavanca o outro. Por fim a convicção de "Play School"como antecedente para "The Fly II"como consequente foi de 1,00070 enquanto de "The Fly II"como antecedente para "Play School"como consequente

foi de 1,02372 mostrando que existe uma dependência entre esses dois itens, porém ela é bem fraca.

Analisando os resultados dos primeiros itens além de "Play School"obtivemos o filme "Homam"e o jogo "Democracy 3"com o suporte em 0,1% entre os dois itens, ou seja, de 50000 pessoas analisadas 50 consumiram os dois itens juntos. A confiança de "Homam"como antecedente para "Democracy 3"como consequente foi de 7,65% e de "Democracy 3"como antecedente para "Homam"como consequente foi de 1,31% valores mais próximos se comparado aos exemplos anteriores para 50000. O lift foi de 1,40331 mostrando que os itens têm uma boa alavancagem entre si maior que o caso anterior. E a convicção de "Homam"como antecedente para "Democracy 3"como consequente 3 foi de 1,02379 enquanto de "Democracy 3"como antecedente para "Homam"como consequente foi de 1,00537 mostrando uma dependência maior que o exemplo anterior, porém ainda baixa.

Por fim, pegando o primeiro exemplo de recomendação para livro "Lectures, addresses"e o filme "Monsters vs Aliens"onde obtivemos 0,106% de suporte entre esses dois itens, ou seja, de 50000 pessoas 53 viram o filme "Monsters vs Aliens"e leram o livro "Lectures, addresses". A confiança de "Lectures, addresses"como antecedente para "Monsters vs Aliens"como consequente foi de 2,06% e de "Monsters vs Aliens"como antecedente para "Lectures, addresses"como consequente foi de 6,65% resultando em valores mais próximos que os itens anteriores. O lift foi de 1,29326 para os dois itens mostrando uma boa alavancagem entre si. Porém pior que o exemplo anterior e o primeiro exemplo. E a convicção de "Lectures, addresses"como antecendete para "Monsters vs Aliens"como consequente foi de 1,00477 enquanto de "Monsters vs Aliens"como antecedente para "Lectures, addresses"como consequente foi de 1,01615. Resultado em uma baixa dependência desses itens obtendo os piores valores até o momento.

4.2.3 Resultados para os 100000 primeiros usuários.

Avançando para os 100000 primeiros resultados e usando o suporte mínimo como 0,001. Obtivemos 1455878 resultados um pouco menos que rodando para 50000 devido a esse exemplo considerar 100 relacionamentos entre dois itens como o mínimo válido. Analisando alguns resultados obtivemos os seguintes casos mostrados na tabela 7.

Antecedents	Consequents	Antecedent Support	Consequent Support	Support	Confidence	Lift	Conviction
The Beverly Hillbillies serie	Play School serie	0.00359	0.20275	0.001	0.27855	1.37387	1.10507
Play School serie	The Beverly Hillbillies serie	0.20275	0.00359	0.001	0.00493	1.37387	1.00135
Play School serie	The Fly II filme	0.20275	0.0075	0.00175	0.00863	1.15084	1.00114
The Fly II filme	Play School serie	0.0075	0.20275	0.00175	0.23333	1.15084	1.03989
The Greatest Game Ever Played filme	Hunter Street serie	0.01317	0.06824	0.00106	0.01553	1.17945	1.0024
Hunter Street serie	The Greatest Game Ever Played filme	0.06824	0.01317	0.00106	0.08049	1.17945	1.01332
Monsters vs Aliens filme	Just Cause™ 3 jogo	0.01588	0.05279	0.00102	0.06423	1.21674	1.01223
Just Cause™ 3 jogo	Monsters vs Aliens filme	0.05279	0.01588	0.00102	0.01932	1.21674	1.00351
Miscellaneous writings livro	Cirque du Soleil: Corteo filme	0.05062	0.01441	0.001	0.01976	1.37093	1.00545
Cirque du Soleil: Corteo filme	Miscellaneous writings livro	0.01441	0.05062	0.001	0.0694	1.37093	1.02018

Tabela 7 – Resultados para 100000 itens

Analisando os resultados obtivemos assim como para 50000 as séries "The Beverly Hillbillies"e "Play School"como os primeiros resultados com o suporte em 0,1% dessa vez significando que das 100000 pessoas analisadas 100 assistiram a essas duas séries. A confiança de "The Beverly Hillbillies"como antecedente para "Play School"como consequente foi de 27,86% enquanto de "Play School"como antecedente para "The Beverly Hillbillies"como consequente foi de apenas 0,49%. Assim como no anterior isso ocorreu devido a "Play School"estar em grande parte dos usuários nesse caso estando em 20,28% das pessoas analisadas. O Lift para esses dois casos foi de 1,37387 mostrando uma boa alavancagem dos dois itens. E a convicção de "The Beverly Hillbillies"como antecedente para "Play

School"como consequente sendo em 1,10507 e de "Play School"como antecedente para "The Beverly Hillbillies"como consequente sendo de 1,00135 mostrando que existe uma dependência dos dois itens, porém ela é baixa. Esses resultados ficam próximos aos resultados do *dataset* anterior trazendo uma credibilidade maior entre os resultados.

Chegando nos primeiros resultados de mídias diferentes novamente obtemos o filme "The Fly II"com a Série "Play School"tendo o suporte entre os dois de 0,18% indicando que das 100000 pessoas analisadas 175 consumiram os dois itens juntos. A confiança de "Play School"como antecedente para "The Fly II"como consequente foi de 0,86% enquanto de "The Fly II"como antecedente para "Play School"como consequente foi de 23,33% o que assim como no *dataset* anterior é causado devido a "Play School"ter uma grande presença no *dataset*. O Lift foi de 1,15084 indicando uma dependência dos dois itens até maior que no *dataset* anterior. E a Convicção de "Play School"como antecedente para "The Fly II"como consequente foi de 1,00114 enquanto de "The Fly II"como antecedente para "Play School"como consequente foi de 1,03989 o que indica assim como exemplo anterior uma dependência fraca mesmo os valores tendo aumentado se comparado ao *dataset* anterior.

Chegando ao primeiro item após os resultados de "Play School"obtivemos o filme "The Greatest Game Ever Played"e a série "Hunter Street"com o suporte em 0,106%, ou seja, das 100000 pessoas analisadas 106 consumiram esses dois itens juntos. A confiança de "The Greatest Game Ever Played"como antecedente para "Hunter Street"como consequente foi de 1,55% enquanto de "Hunter Street"como antecedente para "The Greatest Game Ever Played"como consequente foi de 8,05% mostrando valores mais próximos entre os dois itens devido à presença na base de dados deles serem mais próximas. O Lift foi de 1,17945 mostrando que existe uma alavancagem dos dois itens. E a convicção de "The Greatest Game Ever Played"como antecedente para "Hunter Street"como consequente foi de 1,0024 enquanto de "Hunter Street"como antecedente para "The Greatest Game Ever Played"como consequente foi de 1,01332 indicando assim como no exemplo anterior uma dependência baixa desses dois itens.

Avançando para o primeiro jogo temos o Jogo "Just Cause™ 3"e o filme "Monsters vs Aliens"com 0,102% de suporte para os dois itens, ou seja, das 100000 pessoas analisadas 102 consumiram esses dois itens juntos. A confiança de "Monsters vs Aliens"como antecedente para "Just Cause™ 3"como consequente foi de 6,42% enquanto de "Just Cause™ 3"como antecedente para "Monsters vs Aliens"como consequente foi de 1,93% mostrando valores mais próximos que o exemplo anterior. O Lift é de 1,21674 indicando uma boa alavancagem entre os dois itens e a Convicção de "Monsters vs Aliens"como antecedente para "Just Cause™ 3"como consequente foi de 1,01223 enquanto de "Just Cause™ 3"como antecedente para "Monsters vs Aliens"como consequente foi de 1,00351 o que assim como nos exemplos anteriores indica uma dependência baixa entre os dois itens.

Por fim, avançando para o primeiro livro obtemos a relação entre o livro "Miscellaneous writings"e o filme "Cirque du Soleil: Corteo"com o suporte entre os dois itens em 0,1%, ou seja, das 100000 pessoas analisadas 100 consumiram esses dois itens juntos. A confiança de "Miscellaneous writings"como antecedente para "Cirque du Soleil: Corteo"como consequente foi de 1,98% enquanto "Cirque du Soleil: Corteo"como antecedente para "Miscellaneous writings"como consequente foi de 6,94%. O Lift foi de 1,37093 indicando que existe uma boa alavancagem entre os dois itens. E a Convicção de "Miscellaneous writings"como antecedente para "Cirque du Soleil: Corteo"como consequente foi de 1,00545 enquanto "Cirque du Soleil: Corteo"como antecedente para "Miscellaneous writings"como consequente foi de 1,02018, indicando uma baixa dependência entre os dois itens.

5 Conclusões e Discussões

Esse trabalho aplicou o algoritmo Apriori para um modelo de base de dados que agrega quatro base de dados sobre mídias de entretenimento em uma única. De forma a conseguir gerar recomendações de conteúdos dessas quatro bases interligadas a partir da combinação desses dados.

Por se tratar de bases de dados diferentes a escolha do Apriori para implementação foi bem assertiva uma vez que esse algoritmo realiza todas as comparações possíveis entre os itens buscando relacionamentos, e por se tratar de uma junção experimental essa técnica se encontrou como a melhor para essa ocasião devido à natureza do problema apresentado nesse projeto.

Os resultados desse projeto contribuem como uma forma de gerar recomendações para bases de dados diferentes podendo ser aplicado a diversos contextos e serviços diferentes buscando encontrar relacionamentos para recomendação de conteúdo, como o caso de bases de dados não relacionais como foi o caso desse projeto e outros Big Data.

Ao analisar os resultados obtidos com o algoritmo, mostra-se que ele é capaz de gerar recomendações de mídias diferentes seguindo esse modelo de dados. Grande parte dos resultados apresentaram números aceitáveis para recomendação como um Lift e Conviction maiores que 1. Os valores de suporte e confiança nesse projeto podem ser considerados baixos para os padrões devido a máscara dos dados reais, porém são aceitáveis levando em consideração o fator da base de usuários ter sido criada com itens aleatórios.

Alguns resultados foram um pouco atípicos, como o caso da Serie Play School, que devido a estar presente na maioria das transações analisadas resultando nesse item altamente recomendado em comparação a outros itens. Esse resultado indica uma recomendação para itens mais famosos ou em alta que serão consumidos por muitas pessoas em um meio fortalecendo a sua recomendação.

O algoritmo foi capaz por meio das interações dos usuários gerarem recomendações entre itens em mídias diferentes como mostrado nos *datasets* a aparição com resultados aceitáveis validando o seu funcionamento para aplicação em uma massa maior.

Para trabalhos futuros, busco a implementação de uma aplicação real onde pessoas possam inserir seus dados de consumo e seguindo o modelo desse trabalho. Possa gerar recomendações baseadas em um modelo de dados real, integrando com outros serviços até utilizados nesse trabalho para obtenção das mídias de entretenimento facilitando a obtenção de dados para análise.

A melhoria no desempenho e otimização de uso de recursos. Por se tratar do uso do Apriori que é um dos algoritmos mais custosos e ele ser aplicado dentro de uma base grande de dados. O algoritmo enfrenta grandes problemas de desempenho, alto uso de recursos computacionais e um grande tempo de resposta, o tornando pouco viável para aplicação em um *big data*.

Referências

AGRAWAL, R.; SRIKANT, R. *Fast algorithms for mining association rules*. Santiago de Chile: Proc. of the Int'l Conf. on Very Large Databases, 1994.

CASS, S. *Top Programming Languages 2022, Python's still No. 1, but employers love to see SQL skills*. 2022. Disponível em: <https://spectrum.ieee.org/top-programming-languages-2022>. Acessado em: 20/05/2023.

CASTRO, L. N. d.; FERRARI, D. G. *Introdução à mineração de dados: conceitos básicos, algoritmos e aplicações*. Saraiva, 2016. 26, 27 p. Disponível em: <https://pergamum.mackenzie.br/acervo/5119422/referencia>.

DATE, C. J. *Introdução a sistemas de bancos de dados*. [S.l.]: Elsevier, 2004.

DOCS, G. *about-repositories*. 2024. Disponível em: <https://docs.github.com/pt/repositories/creating-and-managing-repositories/about-repositories>. Acessado em: 20/05/2023.

FERREIRA, A. G. *Interface de programação de aplicações (API) e web services*. Saraiva, 2021. Disponível em: <https://pergamum.mackenzie.br/acervo/5232061/referencia>.

FRERY, A. C. *A Badging System for Reproducibility and Replicability in Remote Sensing Research*. IEEE, 2020. Disponível em: <https://ieeexplore.ieee.org/abstract/document/9177276>.

GITHUB. *about-projects*. 2024. Disponível em: <https://docs.github.com/pt/issues/planning-and-tracking-with-projects/learning-about-projects/about-projects>. Acessado em: 20/05/2023.

LIVRO, C. brasileira do. *O que é o ISBN?* 2023. Disponível em: <https://www.cblservicos.org.br/isbn/o-que-e-isbn/>. Acessado em: 04/11/2023.

OPENLIBRARY. *Centro do Desenvolvedor*. 2021. Disponível em: <https://openlibrary.org/developers>. Acessado em: 20/05/2023.

PIZZI, L. C. *Mineração multi-relacional: o algoritmo GFP-growth*. Universidade Federal de São Carlos, 2006. Disponível em: <https://repositorio.ufscar.br/handle/ufscar/332>.

SADALAGE, P. J.; FOWLER, M. *NoSQL essencial: um guia conciso para o mundo emergente de persistência poliglota*. [S.l.]: Novatec, 2013.

SILVA, L. A. d.; PERES, S. M. *Introdução à mineração de dados: com aplicações em R*. SBC (Sociedade Brasileira de Computação), 2015. Disponível em: <https://pergamum.mackenzie.br/acervo/5143547/referencia>.

SILVA, L. F. C. *Banco de dados não relacional*. SAGAH, 2021. Disponível em: <https://pergamum.mackenzie.br/acervo/5186279>.

THEMOVIEDB. *FAQ*. 2023. Disponível em: <https://developer.themoviedb.org/docs/faq>. Acessado em: 20/05/2023.

VALVE. *Termos de uso da Steam Web API*. 2010. Disponível em: <https://steamcommunity.com/dev/apiterms>. Acessado em: 20/05/2023.