PESQUISA DESCRITIVA SOBRE METODOS DE RECOMENDAÇÃO EM PLATAFORMAS DE STREAMING

Gustavo Vilela Mitraud (IC) e Leandro Carlos Fernandes (Orientador)

Apoio:PIVIC Mackenzie

RESUMO

A presente pesquisa objetivou analisar como um sistema de recomendação utiliza-se de algoritmos de IA e métodos de pré-processamento de dados no contexto de uma plataforma de *streaming* para criar recomendações baseados nos gostos de um usuário. Para fazê-lo, primeiramente mapeamos o fluxograma do processo, identificando e descrevendo cada uma de suas etapas, a saber: Filtragem de dados, Pré-processamento de dados, Aplicação de algoritmos, Avaliação do modelo e Implantação do sistema de recomendação. Em cada um desses passos, discutimos as possíveis alternativas apresentadas pela literatura utilizada em nosso referencial teórico e sua aplicabilidade ao caso concreto. Metodologicamente, aplicamos diferentes modelos a um conjunto de dados retirados da plataforma de venda de jogos Steam. O objeto de estudo específico foi o jogo “Counter-Strike: Global Offensive”, que serviu de *input* para o sistema de geração das recomendações, consideradas como o *output*. Tais recomendações foram construídas a partir de duas abordagens: a descrição textual dos jogos e pelo agrupamento de categorias de cada item dos jogos. Realizados os testes e comparada à similaridade entre o *input* e *outputs* em cada um desses princípios, nossa pesquisa concluiu que a abordagem baseada em um agrupamento das diferentes categorias de cada jogo somado ao um algoritmo de clusterização como o *k-means* é a melhor forma de se aplicar um sistema de recomendação ao nosso caso.

Palavras-chave: *Streaming*. sistemas de recomendação. Inteligência artificial.

ABSTRACT

This research aimed to analyze how a recommendation system uses AI algorithms and data pre-processing methods in the context of a streaming platform to create recommendations based on a user's tastes. To do this, we first mapped out the process flowchart, identifying and describing each of its stages, namely: Data filtering, Data pre-processing, Applying algorithms, Evaluating the model and Deploying the recommendation system. In each of these steps, we discuss the possible alternatives presented by the literature used in our theoretical framework and their applicability to the specific case. Methodologically, we applied different models to a

set of data taken from the Steam games sales platform. The specific object of study was the game "Counter-Strike: Global Offensive", taken as our input to the system for generating recommendations, considered to be the output. These recommendations were constructed using two approaches: the textual description of the games and the grouping of categories for each item in the games. After carrying out the tests and comparing the similarity between the input and outputs in each of these principles, our research concluded that the approach based on a grouping of the different categories of each game added to a clustering algorithm such as k-means is the best way to apply a recommendation system to our case.

Keywords: *Streaming*. *Recommendation Systems. Artificial Intelligence*

1. INTRODUÇÃO

Em 2023, o patrimônio de Daniel Elk, cofundador e atual CEO do Spotify, foi estimado em 2,5 Bilhões de dólares (FORBES, 2023), sendo está a plataforma de *streaming* de música a principal forma que ouvintes do mundo todo apreciam seus artistas favoritos. As plataformas de *streaming* e *e-commerce* dominam não apenas o mundo da música, mas também uma grande parte das atividades da sociedade contemporânea, sejam elas culturais, de lazer, compra e venda de mercadorias e diversas outras. Netflix, HBO, Disney+ e Paramount são outras gigantes destes segmentos que cresceram muito rapidamente, deixando seus acionistas contentes com os resultados, como Jeff Bezos, fundador da gigante da tecnologia Amazon, que em 2023 foi considerado o terceiro homem mais rico do mundo (FORBES, 2023).

Há algo que une todas essas gigantes empresas de tecnologia as quais enriquecem seus criadores: a utilização de sistemas de recomendação baseados em modelos de IA (inteligência artificial). Seja em recomendar artistas novos ou produtos que são frequentemente vendidos juntos, as inteligências artificiais são utilizadas em peso por empresas cujo sucesso é reconhecido por todos.

Em uma publicação para o jornal da ACM (*Association for Computing Machinery*) em 1997, Paul Resnick e Hal R. Varian citam que um sistema de recomendação é capaz de auxiliar e ampliar o processo natural de *“word of mouth”* ou seja, a recomendação por “boca a boca”. Contudo, em casos em que a quantidade de informação e de usuários é muito grande, utilizar uma recomendação de uma pessoa se torna inviável. A grande quantidade de dados gerados e coletados no mundo moderno, o aumento exponencial no número de usuários e a enorme competitividade entre essas empresas de *streaming* e *e-commerce* faz com que a procura por uma forma de recomendar produtos e serviços para um usuário de forma autônoma, ou seja, usando um modelo de IA, se torne grande e constante.

O objetivo geral da pesquisa é, então, entender como funciona o processo de recomendação para o usuário. Assim sendo, este trabalho propõe responder a seguinte pergunta-problema: como funciona um sistema de recomendação em plataformas de *streaming?* Ademais, foram adotados como objetivos específicos pesquisar como ocorre o pré-processamento dos dados utilizados numa recomendação e estudar quais são os algoritmos de IA utilizados.

Para a análise e condução desse estudo, foi utilizada a plataforma de *streaming* de jogos digitais Steam. Criada pela gigante do mundo dos jogos Valve em 2003 (SAYER;

WILDE, 2022), a plataforma rapidamente cresceu e hoje conta com mais 50 mil jogos (WISE, 2023) e até 31 milhões de jogadores ao mesmo tempo (STEAM, 2023).

2. REFERENCIAL TEÓRICO

Antes de tudo, é necessário explicar qual é a cadeia de processos de um sistema de recomendação.

Figura 1: Fluxograma de sistema de recomendação

Fonte: elaborado pelo autor.

O processo de filtragem dos dados consiste em receber os dados recolhidos dos usuários e produtos e filtrá-los para que sejam utilizados no restante do processo. Esta fase é importante para a criação do modelo de recomendação, pois para um específico problema, nem todos os dados coletados são pertinentes de serem utilizados.

A fase de pré-processamento dos dados constitui-se no tratamento dos dados já filtrados com o intuito de otimizar a aplicação do algoritmo de IA. Esse tratamento pode se dar de várias formas, como, por exemplo: calcular a similaridade entre textos e transformar dados textuais em numéricos (CAZELLA; NUNES; REATEGUI, 2010), e redimensionar os dados, passo importante para a otimização do tempo de execução da recomendação e para certos algoritmos de IA, como os SVM (*Support Vector Machine*) (MULLER; GUIDO, 2016). Este passo é de vital importância para qualquer sistema de recomendação, visto que os dados recolhidos dos usuários e produtos nem sempre estarão no formato perfeito para a IA.

A aplicação do algoritmo de IA é um processo complexo, que pode ser divido em várias partes. Dentre elas: Escolha do algoritmo, Separação da base de dados em treinamento e teste, treinamento do modelo, e otimização dos parâmetros (MULLER; GUIDO, 2016). A escolha do algoritmo refere-se a identificar qual é o melhor algoritmo para um certo problema específico. A separação da base de dados ocorre apenas em problemas que demandam IAs supervisionadas, conforme descrito na seção 2.2 a seguir, e consiste em separar os dados para que uma parte seja utilizada para treinar o modelo e a outra parte seja utilizada para validação. O treinamento do modelo abrange a alimentação dos dados no algoritmo de IA para a criação do modelo. Por fim, a otimização dos parâmetros, a qual ocorre principalmente

após uma avaliação inicial do modelo, consiste na mudança dos parâmetros da IA para melhorar a performance do modelo (Idem, 2016).

A avaliação do modelo pode ser feita de diversas formas, cada uma dependendo do algoritmo de IA utilizado, como, por exemplo, WCSS para o *k-means clustering* (MACQUEEN, 1967) e a mAP para algoritmos de detecção de objetos (SHAH, 2022). Algo a se notar nessas avaliações de cunho matemático, é que nem sempre elas são as perfeitas formas de se avaliar um modelo, pois mesmo quando se tem uma performance boa nessas métricas, a performance real ao usuário pode ser pior que o esperado.

2.1 Métodos de filtragem

Antes da realização do modelo de IA em si, é necessário entender quais são as diferentes formas de filtrar os dados. A chamada "filtragem colaborativa" é a mais antiga abordagem, sendo utilizada no primeiro sistema de recomendação *Tapestry* (GOLDBERG et al., 1992; CAZELL.; NUNES; REATEGUI, 2010), ela consiste na associação de grupos interessados em um mesmo produto ou serviço, por exemplo: se um usuário A gosta dos produtos x_1, x_2, x_3 ... x_n e um produto y, então B, que também gosta dos produtos x_1, x_2,x_3 ... x_n, provavelmente também irá gostar do produto y (BEEL et al., 2016). Embora efetiva, esta abordagem tem um grande problema: como recomendar para um usuário novo que não pertence a nenhum grupo? Ou quando um novo produto é adicionado, não possuindo compras e avaliações suficientes, como o recomendamos? Este problema, denominado *cold-start*[1], fez com que o uso puro da "filtragem colaborativa" tenha caído em desuso, sendo, hoje em dia, mais utilizada em modelos híbridos, ou seja, que usam mais de um tipo de filtragem (Idem, 2016).

A filtragem mais utilizada e pesquisada hoje em dia é a "filtragem baseada em conteúdo", mais comumente chamada por seu nome em inglês: "*Content-based filtering*" ou CBF (DAUD; SHAIKH; RAJPAR, 2009, p. 255). Ela utiliza o fato de que produtos com algumas características parecidas são, em geral, semelhantes, ou seja, podem ser recomendados juntos. Uma das formas mais comuns em que a CBF é aplicada é no uso de "*tags*", "*flags*" ou "*tokens*". Está forma de aplicar consiste em dar a um certo produto certas palavras-chave que serão compartilhadas entre outros parecidos. O uso dessas palavras-chave pode ser visto em sistemas de recomendação de artigos científicos, que se utilizam de palavras-chave retiradas do título e do corpo do texto (BEEL et al., 2016).

---

[1] Essa expressão que, em tradução livre, significa *início frio*, é usada quando um sistema de recomendação não possui previamente dados suficientes para fazer uma sugestão.

2.2 Algoritmos de IA

Falando agora dos algoritmos de IA que são utilizados em sistemas de recomendação, existem duas grandes categorias: algoritmos supervisionados e não-supervisionados (MULLER; GUIDO, 2016). Algoritmos supervisionados são utilizados geralmente em problemas de classificação, como, por exemplo, determinar a espécie de uma flor pelas características de suas pétalas. Para funcionar, eles precisam de dados que já foram classificados previamente, o modelo então é "treinado" nesses dados, podendo ser aplicados, assim, em dados não classificados (MULLER; GUIDO, 2016).

Já algoritmos não-supervisionados são utilizados em problemas de "clusterização" (agrupamento de dados similares), que é o caso deste trabalho. Esses algoritmos de IA, diferentemente dos supervisionados, não utilizam dados já previamente rotulados e sim a aplicação de funções matemáticas nas categorias de cada elemento do conjunto de dados para descobrir padrões (MULLER; GUIDO, 2016).

Um dos mais famosos e comuns desses algoritmos é o *K-Means Clustering.* Descrito por MacQueen (1967), ele consiste, de forma simplificada, em randomicamente selecionar $k \in \mathbb{N}$ pontos do conjunto de dados iniciais, estes são chamados de centroides. Então cada ponto é atribuído a um centroide tal que a distância Euclidiana (também é possível, como descrito por Singh, Yadav e Rana (2013), utilizar outras métricas de distância) seja minimizada, ou seja (MACQUEEN, 1967):

$$k^i = \min \| X_i - Y_j \|^2$$

Sendo k^i o centroide que o ponto X_i é atribuído e Y_i a posição de cada centroide.

Após todos os pontos originais serem atribuídos a um centroide, a posição dos centroides é atualizada para ser a média de todos os pontos atribuídos a ele. Isto se repete até o número de iterações pré-definidas, ou se a posição dos centroides não mudar.

Isto faz com que naturalmente grandes grupos de pontos se formem e que os elementos do banco de dados utilizado sejam distribuídos de forma que os pontos mais parecidos estejam próximos uns aos outros. O foco do algoritmo é minimizar a chamada *within-cluster sum of squares* (WCSS). Quando a métrica de distância utilizada é a Euclidiana, ela é dada pela função (MACQUEEN, 1967):

$$WCSS = \sum_{j=1}^{k} \sum_{i \in S_j} \left\| X_i - Y_j \right\|^2$$

onde S_j é o conjunto de pontos atribuídos ao centroide Y_j.

Ou seja, a soma das distâncias ao quadrado dos pontos X_i em relação a posição do centroide Y_j deve ser mínima.

Finalizando, para determinar a quantidade ideal de *clusters* em uma aplicação do *kmeans*¸ podem ser utilizadas diversas métricas: a mais famosa e simples dentre elas é o conhecido como método do cotovelo, que consiste em, primeiramente, calcular a WCSS para um conjunto de número de *clusters* de teste (o método do cotovelo é indicado para um número pequeno de *clusters*, como descrito por Cui (s.d.), então constrói-se um gráfico onde o eixo x são as quantidades de *clusters* e o eixo y são suas respectivas WCSS. Por fim, verifica-se em qual ponto do eixo x há uma queda dramática no valor do eixo y, aquele será o valor ideal de *clusters* (UMARGONO, E.; SUSENO, J. E.; GUNAWAN, S. K. V, 2019).

2.3. Métricas de similaridade

Ao tentar resolver um problema de recomendação é muito comum encontrar dados textuais, como descrições de jogos, que terão que ser comparados para verificar sua semelhança. Em baixa escala, uma pessoa facilmente realiza esta ação, porém, ao aumentar a quantidade de dados e operações necessárias para uma recomendação, encontra-se inviável utilizar um humano para calcular essa semelhança.

Portanto, é necessário encontrar uma forma de transformar estes dados textuais em numéricos, visto que a linguagem humana não pode ser compreendida pelas máquinas e por algoritmos matemáticos. Uma forma de realizar esta transformação é aplicar o chamado TFIDF (*term frequency-inverse document frequency*)[2] em conjunto com uma métrica de distância ou algoritmo de clusterização, como descrito por Kim e Gil (2019).

O TF-IDF consiste na junção de dois valores calculados para um certo termo: o TF (*term frequency*), ou a frequência daquele termo dentro de um texto, e o IDF (*inverse document frequency*), ou o inverso da quantidade de vezes que um texto (no caso deste trabalho, descrições de jogos) possui aquele termo.

Matematicamente, define-se o TF como (KIM; GIL, 2019):

$$TF_{i,j} = \frac{n_{i,j}}{M_j}$$

Onde $n_{i,j}$ é a quantidade de vezes que a palavra i aparece no texto j e M_j é a quantidade de palavras no documento j. Ou seja, será feita uma varredura em todo o texto e calcular o TF para cada palavra em cada texto. Valores altos de TF para uma palavra em um certo texto indicam que este termo aparece muitas vezes naquele específico texto.

[2] Em tradução livre, frequência de termo-frequência inversa de documento

Já o IDF, é definido, matematicamente (KIM; GIL, 2019), como:

$$IDF_i = \log\left(\frac{|D|}{|j \in D : i \in j|}\right)$$

Onde D é o conjunto de todos os documentos e $j \in D : i \in j$ é o conjunto dos documentos onde a palavra i aparece. Um valor alto de IDF para um termo qualquer significam que esta palavra é rara no conjunto de todos os documentos.

Por fim, o TFIDF é estabelecido como a multiplicação dessas duas operações (o TF e o IDF). Portanto, matematicamente (KIM; GIL, 2019):

$$TFIDF = TF \cdot IDF$$

Cria-se então, uma matriz com o valor de $TFIDF$ para cada palavra em relação a cada documento:

Quadro 1. Matriz de TFIDF

	Palavra 1	Palavra 2	Palavra 3	...	Palavra n
Texto 1	$TF_{1,1} \cdot IDF_1$	$TF_{2,1} \cdot IDF_2$	$TF_{3,1} \cdot IDF_3$	...	$TF_{n,1} \cdot IDF_n$
Texto 2	$TF_{1,2} \cdot IDF_1$	$TF_{2,2} \cdot IDF_2$	$TF_{3,2} \cdot IDF_3$	...	$TF_{n,2} \cdot IDF_n$
Texto 3	$TF_{1,3} \cdot IDF_1$	$TF_{2,3} \cdot IDF_2$	$TF_{3,3} \cdot IDF_3$	...	$TF_{n,3} \cdot IDF_n$
⋮	⋮	⋮	⋮	⋱	⋮
Texto m	$TF_{1,m} \cdot IDF_1$	$TF_{2,m} \cdot IDF_2$	$TF_{3,m} \cdot IDF_3$	...	$TF_{n,m} \cdot IDF_n$

Fonte: elaborado pelo autor.

Com $n = numero\ total\ de\ palavras\ distintas\ em\ todos\ os\ documentos$ e $m = numero\ de\ documentos$.

Tendo em vista esta matriz, podemos dizer que textos que possuem valores parecidos em suas colunas (os valores de $TFIDF$) são em parte parecidos, pois esses valores indicam que estes dois textos tem palavras que aparecem apenas neles, e não em outros documentos.

Após a realização do algoritmo, podem ser aplicadas diversas formas de calcular a similaridade (ou a distância entre cada texto dentro da matriz do $TDIDF$). A utilizada neste

trabalho, no contexto deste algoritmo, é a similaridade de cosseno, definida matematicamente como (RAHUTOMO, F.; KITASUKA, T.; ARITSUGI, M., 2012):

$$sim(\vec{n}, \vec{m}) = \frac{\vec{n} \cdot \vec{m}}{\|n\| \cdot \|m\|}$$

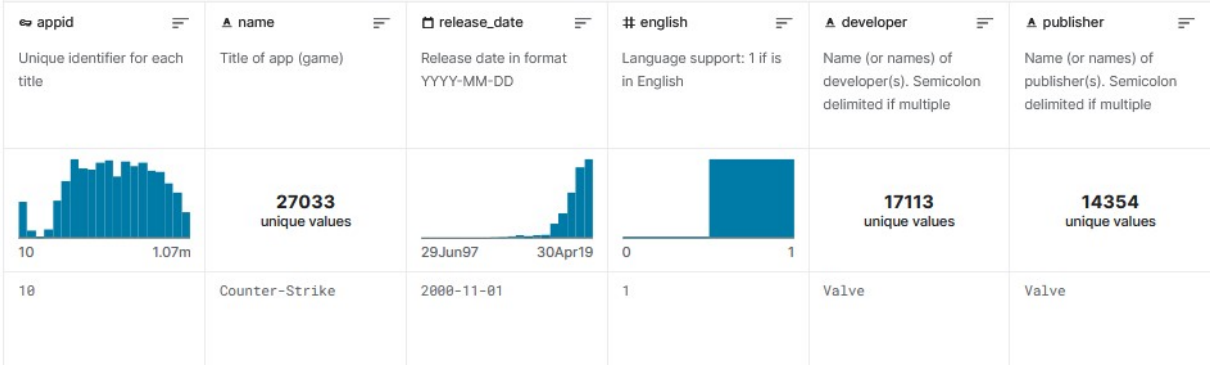
sendo $\vec{n}$ e $\vec{m}$ vetores de características (as colunas da matriz criada pelo $TFIDF$).

3. METODOLOGIA

Ao longo do trabalho, foram testados 3 modelos de recomendação, em ordem crescente de desempenho. A linguagem de programação utilizada na realização desses modelos foi o Python aplicado no Jupyter Notebook, com a adição da biblioteca Pandas para a leitura e processamento dos dados; Numpy para operações matemáticas com *arrays,* e Scikit-Learn para a aplicação dos modelos de *clusterização* e a seleção dos jogos, já separados nas diferentes categorias criadas pelo *k-means*, para a recomendação (MITRAUD, 2023).

A base de dados referente a Steam for retirada da comunidade de Machine learning e Data Science Kaggle. Ela consiste em 17 colunas com valores inteiros, texto em inglês, datas e intervalos (DAVIS, 2019); e em uma tabela com a descrição textual em inglês feita pelos desenvolvedores de cada jogo.

Figura 2. Corte do jogo Counter-Strike na base com os dados de cada jogo



Fonte: DAVIS, 2019.

Para o propósito deste trabalho, foram utilizadas principalmente (mas não unicamente) seis das mais importantes, sendo elas:

- ***appid:*** Um número que identifica o jogo, é um valor abstrato que não possui significado numérico;
- ***name:*** O nome do jogo, utilizado principalmente como *input* para a recomendação;
- ***genre:*** Gênero (ação, aventura, ficção cientifica etc.) do jogo
- ***steamspy_tags*****:** Palavras-chave do jogo, dadas pelo site SteamSpy

- ***categories:*** Categorias do jogo. Específica mais detalhadamente o jogo.

Visto que os dados são referentes a jogos e não a usuários, a recomendação será feita recebendo o nome de um jogo como *input* e retornará os jogos recomendados para aquele *input* específico, não havendo interferência de dados dos usuários.

3.1. Métodos de pré-processamento e análise inicial dos dados.

Antes da criação de um modelo em si, foram testadas certas formas de processar os dados da base de dados original para a melhor aplicação do *k-means*. Isto foi feito, pois as categorias do conjunto de dados não só não possuem o mesmo tipo (por exemplo: a coluna *appid* é um número, já as colunas *steamspy_tags* e *categories* são um conjunto de nomes), mas também não são numéricos, ou seja, é impossível aplicar o algoritmo neles, pois o *kmeans* necessita de um conjunto de dados numérico para funcionar.

Portanto, primeiramente, foi feita uma discussão de como seria possível transformar os dados não numéricos em numéricos:

- Para as colunas *genre, steamspy_tags* e *categories*, que são conjuntos de nomes separados por uma vírgula, foi utilizado um algoritmo, que será explicado mais adiante, o qual recebe quais os gêneros, *tags* do SteamSpy e categorias que aquele jogo pertence e retorna grandes grupos dessas características. Sendo possível que jogos parecidos, mas não necessariamente iguais, possam ser recomendados juntos.
- Por fim, para um dos modelos, o qual é baseado apenas na descrição textual dos jogos, foi utilizado uma mistura do TF-IDF (*Term Frequency-Inverse Document Frequency*), um algoritmo que calcula o peso de uma palavra em um conjunto de textos, e da distância de cosseno. Isto nos dá uma matriz que possui a similaridade de cada jogo em relação a cada outro.

3.2. Modelo baseado na descrição textual.

O primeiro modelo foi criado com base na seguinte hipótese: A descrição textual para um certo produto é a melhor forma de descrevê-lo, pois não ocorre a necessária redução do detalhamento quando o reduzimos a características únicas. Portanto, caso dois produtos tenham descrições “parecidas”, eles devem ser parecidos e, então, recomendados.

Assim sendo, foi necessário pensar em uma forma matemática de identificar a similaridade entre textos, sendo possível assim realizar uma recomendação sem ser necessário uma pessoa ler a descrição dos jogos. Foram discutidas algumas abordagens para solucionar este problema: primeiramente, foi pensado em utilizar a frequência de

palavras na descrição de um jogo, e assim criar a chamada matriz de frequência, que contêm a quantidade de vezes que cada palavra aparece em cada texto (neste caso, em cada descrição).

Quadro 2, matriz de frequência

	Palavra 1	Palavra 2	...	Palavra i-1	Palavra i
Jogo 1	$x_{1,1}$	$x_{2,1}$	...	$x_{i-1,1}$	$x_{i,1}$
Jogo 2	$x_{1,2}$	$x_{2,2}$	...	$x_{i-1,2}$	$x_{i,2}$
Jogo 3	$x_{1,3}$	$x_{2,3}$	...	$x_{i-1,3}$	$x_{i,3}$
⋮	⋮	⋮	⋱	⋮	⋮
Jogo j	$x_{1,j}$	$x_{2,j}$	...	$x_{i-1,j}$	$x_{i,j}$

Fonte: elaborado pelo autor.

Sendo $x_{i,j}$; $i, j \in \mathbb{N}$ o número de vezes que a palavra i aparece no texto j.

Após a criação dessa matriz, é possível calcular a distância da descrição de cada jogo em relação a outro e então realizar a recomendação baseada em quais distancias são menores em relação a um certo jogo, ou seja, quais jogos são mais similares.

Um problema, porém, surge ao tentar calcular a distância baseada apenas na frequência das palavras: certas palavras são muito comuns e não adicionam algo de relevante à recomendação. Artigos, preposições e termos como "*game*" não só são extremamente frequentes em qualquer descrição de um produto, mas também podem, caso utilizada apenas a frequência para a recomendação, enviesar o modelo, prejudicando a performance. Para contornar esse problema, algumas formas de ponderar as palavras e extrair de um texto os termos mais importantes existem. Para este trabalho, foi utilizado o $TFIDF$, portanto, termos em inglês como "*the*", *"be*" e *"to",* que aparecem em quase todo texto da língua inglesa (Oxford Dictionaries, 2011) tem um peso muito menor do que palavras que descrevem o jogo em si.

Após a criação da matriz de valores do $TFIDF$, foi calculada a similaridade entre as descrições de cada jogo, criando a matriz tal como dada no Quadro 3.

Quadro 3. Matriz criada após a aplicação da similaridade de cosseno

	Jogo 1	Jogo 2	Jogo 3	...	Jogo M
Jogo 1	1	$sim(1, 2)$	$sim(1,3)$	...	$sim(1,M)$
Jogo 2	$sim(2, 1)$	1	$sim(2,3)$	...	$sim(2,M)$
Jogo 3	$sim(3,1)$	$sim(3,2)$	1	...	$sim(3,M)$
⋮	⋮	⋮	⋮	⋱	⋮
Jogo M	$sim(M,1)$	$sim(M,2)$	$sim(M,3)$	...	1

Fonte: elaborado pelo autor.

sendo $sim\left(x,y\right);x, y \leq M$ a similaridade entre o jogo x e o jogo y.

Após isso, podemos verificar, para realizar a recomendação, quais jogos (colunas) possuem o valor mais próximo de 1 para um certo jogo (linha): esses, de acordo com a hipótese inicial, serão os jogos mais similares e, portanto, os que devem ser recomendados.

3.3. Modelo baseado em “flags” para cada categoria dos jogos.

Para o próximo modelo foi utilizada apenas a coluna *steamspy_tags*, devido ao fato que este modelo foi criado para testar o método de pré-processamento denominado, por este trabalho, de “flags”. Os dados da *steamspy_tags* dispõem-se como apresentado na Figura 3.

Figura 3 *steamspy_tags* para os primeiros 7 jogos da base de dados

appid	name	steamspy_...
10	Counter-Strike	Action;FPS;Multiplayer
20	Team Fortress Classic	Action;FPS;Multiplayer
30	Day of Defeat	FPS;World War II;Multiplayer
40	Deathmatch Classic	Action;FPS;Multiplayer
50	Half-Life: Opposing Force	FPS;Action;Sci-fi
60	Ricochet	Action;FPS;Multiplayer
70	Half-Life	FPS;Classic;Action

Fonte: DAVIS, 2019.

Inicialmente, pergunta-se: como verificar a similaridade das *tags*, ou seja, como dizer que um jogo tem *tags* similares a outro? Considerando que os dados são bem mais

“exatos” (temos um conjunto finito e relativamente pequeno de palavras para o todos os jogos), há uma forma diferente de verificar esta similaridade à proposta no modelo anterior.

Primeiramente, cria-se um conjunto com todas as *tags* distintas no conjunto de todos os jogos, ou seja:

$$T = \{\text{Action, FPS, Multiplayer, } \ldots\}$$

Então, é feita uma matriz cujas colunas representam os elementos de T e as linhas os jogos da base de dados. Caso um jogo possua uma certa *tag*, o elemento na coluna correspondente e na linha do jogo receberá o valor de 1; caso contrário, recebe 0.

Figura 4. Matriz de tags em relação aos jogos, representados pelo seu appid

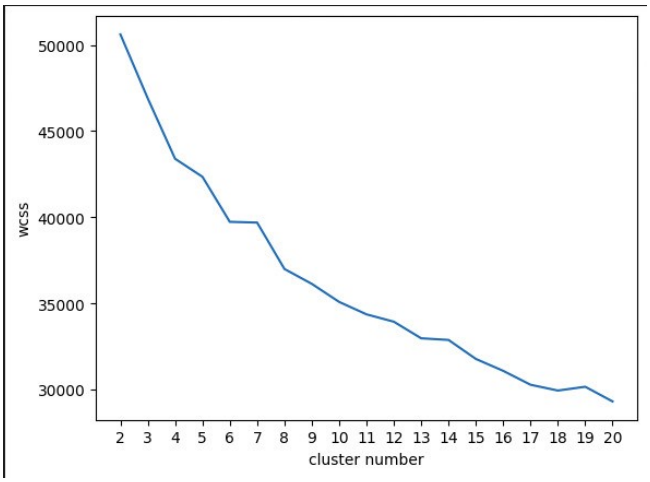
	0	1	2	3	4	5	6	7	8	9	...	329	330	331	332	333	334	335	336	337	338
10	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	0	0	0
40	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1065230	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1065570	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1065650	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1066700	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1069460	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

27075 rows × 339 columns

Fonte: elaborado pelo autor.

É, então, aplicado o *kmeans* com um número de *clusters* de 2 a 20 com o intuito de calcular a WCSS e descobrir o número de *clusters* ideal.

Figura 5. Gráfico de WCSS por quantidade de cluster.



Fonte: elaborado pelo autor.

Por meio do método do cotovelo, descrito no referencial teórico deste trabalho, verifica-se que o número ideal de *clusters* neste caso em específico é de oito.

Por fim, para realizar a recomendação em si, descobre-se em qual *cluster* está o jogo dado como *input*¸ constata-se quais são os 5 jogos com maior semelhança no próprio *cluster*: esses são os que devem ser recomendados.

3.4. Continuação do modelo baseado em flags.

Há um grande problema na forma como foi criada a matriz de *flags* no método anterior: os dados estão extremamente dispersos. Temos 338 colunas e 27075 linhas, as quais, em sua maioria, estão preenchidas com zeros, tendo em vista que os jogos têm, geralmente, de três a quatro *tags* apenas. Quando alimentamos um algoritmo de *clusterização* com dados dispersos desta forma, a distância entre cada elemento da base de dados será grande, pois cada jogo é detalhado demais, prejudicando a performance da recomendação.

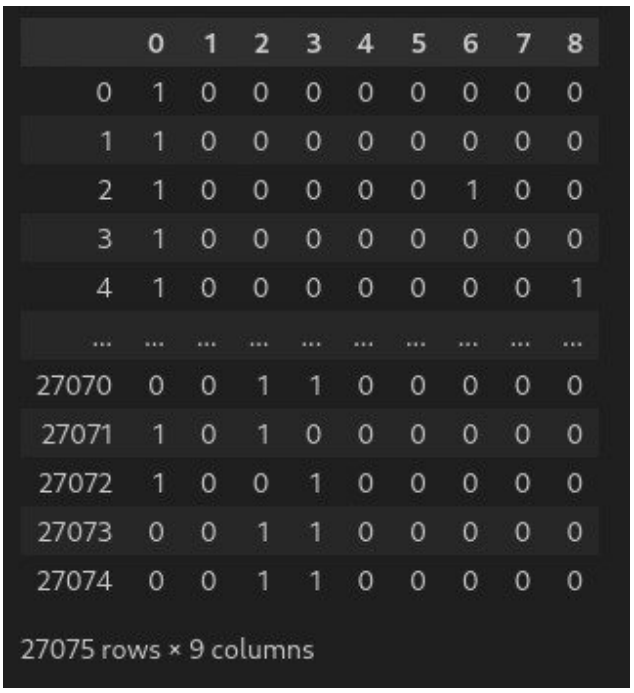
Para contornar este problema, foi proposto o agrupamento de *tags* similares em um conjunto que abstrai a principal ideia daquela *tag*, por exemplo: o conjunto *action* possui *action*, *action-rpg*, *FPS*, *First-Person Shooter* etc. Isto não só agrupa *tags* parecidas, mas também remove as que tem o mesmo significado, como *First-Person Shooter* e *FPS*, o qual é a apenas a abreviação deste.

Foram definidas 9 grandes conjuntos de *tags*3, sendo eles: “*Action*”, “*Role-Playing*”, “*Adventure*”, “*Casual*”, “*Sports & Racing*”, “*Simulation & Strategy*”, “*Historial & Historial Fantasy*”, “*Fictional Fantasy*” e “*Sci-Fi*”. É importante realçar que muitas *tags* fazem parte de mais de um conjunto.

Então é criada uma matriz que segue a mesma filosofia do método anterior: as linhas são os jogos e as colunas os grandes conjuntos de *tags*, caso um jogo possuir uma *tag* que esteja neste conjunto, aquele valor recebe 1, caso o contrário, recebe 0.

[3] A descrição de cada conjunto e quais *tags* fazem parte dele estão localizados no repositório publico do github deste trabalho

Figura 6. Matriz do conjunto de *tags* em relação aos jogos, representados pelo seu appid



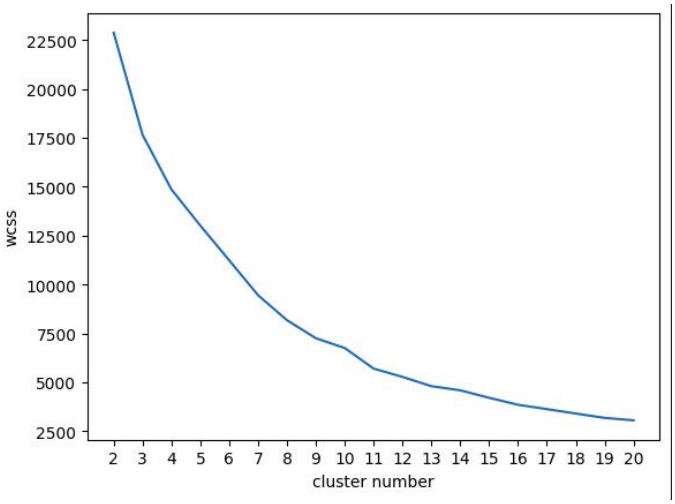
	0	1	2	3	4	5	6	7	8
0	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	1	0	0
3	1	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	1
...	...	...	...	...	...	...	...	...	...
27070	0	0	1	1	0	0	0	0	0
27071	1	0	1	0	0	0	0	0	0
27072	1	0	0	1	0	0	0	0	0
27073	0	0	1	1	0	0	0	0	0
27074	0	0	1	1	0	0	0	0	0

27075 rows × 9 columns

Fonte: elaborado pelo autor.

Como no método anterior, cria-se um gráfico onde o eixo *x* são as quantidades de *clusters* em uma aplicação do *kmeans* e o eixo *y* são suas respectivas WCSS

Figura 7. Gráfico de WCSS por quantidade de cluster.



Fonte: elaborado pelo autor.

É então realizada a recomendação como no método anterior, usando 8 como o número de *clusters*.

4. RESULTADO E DISCUSSÃO

O jogo utilizado como teste para os modelos foi o “Counter-Strike: Global Offensive”, pois ele foi e é o jogo mais popular da Steam, com um máximo de 1802853 jogadores *online* simultaneamente (STEAM CHARTS, 2023).

Para a avaliar o desempenho de cada modelo, serão feitos 5 testes de recomendação, que consistem em inserir o nome de um certo jogo no modelo e receber 5 recomendações, então essas recomendações serão avaliadas para saber se são coerentes com o jogo utilizado como *input*, ou seja, se suas categorias são similares, se fazem parte do mesmo gênero e se tem *tags* do SteamSpy parecidas.

Utilizando o primeiro modelo, baseado na descrição textual, recebemos as seguintes recomendações:

***Input*:**

Figura 8. Categorias, gêneros e tags do SteamSpy do jogo Counter-Strike: Global Offensive

	name	categories	genres	steamspy_tags
25	Counter-Strike: Global Offensive	Multi-player;Steam Achievements;Full controlle...	Action;Free to Play	FPS;Multiplayer;Shooter

Fonte: elaborado pelo autor.

***Output*:**

Figura 9. Resultado da recomendação feita pelo modelo baseado na descrição textual

	name	categories	genres	steamspy_tags
5952	Void And Meddler	Single-player;Steam Achievements;Steam Trading...	Adventure;Indie	Adventure;Cyberpunk;Point & Click
21708	Hyperspace Dogfights	Single-player;Partial Controller Support	Action;Indie	Indie;Action;Arcade
20	Left 4 Dead	Single-player;Multi-player;Co-op;Steam Achieve...	Action	Zombies;Co-op;FPS
10	Counter-Strike: Source	Multi-player;Cross-Platform Multiplayer;Steam ...	Action	Action;FPS;Multiplayer
2506	Descent 3	Single-player;Multi-player	Action	Action;6DOF;FPS

Fonte: elaborado pelo autor.

Interpretando os resultados, podemos ver que, embora os gêneros de cada jogo sejam parecidos (4 dos 5 jogos possuem o gênero *Action*), as categorias e as *tags* do SteamSpy não são similares o suficiente para que a recomendação seja coerente, ou seja, a única *tag* do *input* original que aparece com frequência nos *outputs* é “*FPS”*.

Isto vai contra a hipótese original de que a recomendação seria suficientemente coerente. Um dos possíveis motivos para isso é que nem sempre a descrição dada por um desenvolvedor ao seu jogo é o melhor descritivo para este, pois essas variam muito em sua clareza de jogo a jogo. Tomando como exemplo os jogos “*Dark Souls III”* e “*Elden Ring*”, feitos pelo mesmo desenvolvedor (“*FromSoftware*”) e que possuem *tags* extremamente parecidas, ambos possuem: “*Souls-Like*”, “*Dark Fantasy*”, “*Difficult*”, “*RPG*”, “*Action RPG*” e *“Open World*” (DAVIS, 2019). É evidente que estes jogos devem com certeza serem recomendados juntos,

porém, quando verificamos a descrição deles (“*Elden Ring*” e “*Dark Souls III*”, respectivamente):

> “*THE NEW FANTASY ACTION RPG. Rise, Tarnished, and be guided by grace to brandish the power of the Elden Ring and become an Elden Lord in the Lands Between.”* (STEAM, 2022).
>
> “*Dark Souls continues to push the boundaries with the latest, ambitious chapter in the critically-acclaimed and genre-defining series. Prepare yourself and Embrace The Darkness!”* (STEAM, 2016).

Verifica-se que elas não são similares, contendo muito jargão de *marketing.* Portanto, constatou-se que criar um sistema de recomendação apenas utilizando-se da descrição textual de um item é equivocado, já que esta descrição não irá necessariamente descrever um produto, podendo conter informações desnecessárias para uma recomendação.

Empregando o segundo modelo, temos o seguinte resultado:

***Input*:**

Figura 10. Categorias, gêneros e tags do SteamSpy do jogo Counter-Strike: Global Offensive

	name	categories	genres	steamspy_tags
25	Counter-Strike: Global Offensive	Multi-player;Steam Achievements;Full controlle...	Action;Free to Play	FPS;Multiplayer;Shooter

Fonte: elaborado pelo autor.

***Output*:**

Figura 11. Resultado da recomendação feita pelo modelo baseado em flags

	name	categories	genres	steamspy_tags
25	Counter-Strike: Global Offensive	Multi-player;Steam Achievements;Full controlle...	Action;Free to Play	[FPS, Multiplayer, Shooter]
60	Bloody Good Time	Single-player;Multi-player;Steam Achievements	Action	[Action, Multiplayer, FPS]
26233	Excive A-1000	Single-player	Action	[Action, FPS, Shooter]
146	Natural Selection 2	Multi-player;Online Multi-Player;Steam Achieve...	Action;Indie;Strategy	[Multiplayer, Strategy, FPS]
1	Team Fortress Classic	Multi-player;Online Multi-Player;Local Multi-P...	Action	[Action, FPS, Multiplayer]

Fonte: elaborado pelo autor.

Interpretando os resultados, percebemos que o modelo teve uma melhora significativa em sua performance, visto que todos os jogos possuem as *tags FPS* ou *Shooter*, algo condizente com o *input*. Porém, como dito anteriormente, os dados utilizados são muito dispersos e caso feita uma redução na dimensão destes, é possível melhorar mais ainda a performance.

Por fim, aplicando o último modelo:

***Input*:**

Figura 6. Categorias, gêneros e tags do SteamSpy do jogo Counter-Strike: Global Offensive

	name	categories	genres	steamspy_tags
25	Counter-Strike: Global Offensive	Multi-player;Steam Achievements;Full controlle...	Action;Free to Play	FPS;Multiplayer;Shooter

Fonte: elaborado pelo autor.

Output:

Figura 7. Resultado da recomendação feita pelo modelo baseado no agrupamento de flags

	name	categories	genres	steamspy_tags
1	Team Fortress Classic	Multi-player;Online Multi-Player;Local Multi-P...	Action	[Action, FPS, Multiplayer]
6	Half-Life	Single-player;Multi-player;Online Multi-Player...	Action	[FPS, Classic, Action]
3	Deathmatch Classic	Multi-player;Online Multi-Player;Local Multi-P...	Action	[Action, FPS, Multiplayer]
0	Counter-Strike	Multi-player;Online Multi-Player;Local Multi-P...	Action	[Action, FPS, Multiplayer]
5	Ricochet	Multi-player;Online Multi-Player;Valve Anti-Ch...	Action	[Action, FPS, Multiplayer]

Fonte: elaborado pelo autor.

Observando os jogos recomendados, percebe-se que o modelo possui uma ótima performance. Todos os jogos, com exceção do *Half-Life*, que é um precursor da série *Counter-Strike* e foi feito pela mesma empresa, não só possuem o mesmo conjunto de *tags*, mas também foram feitos pelo mesmo desenvolvedor.

Podemos assim concluir que a abordagem de criar *flags* para cada categoria distinta de um banco de dados e agrupá-las manualmente com base em sua similaridade é uma estratégia que pode ser aplicada com eficácia para dados similares aos utilizados neste trabalho.

5. CONSIDERAÇÕES FINAIS

Este trabalho dedicou-se a responder como um sistema de recomendação pode funcionar em um contexto de uma plataforma de streaming. De tal forma, foram feitas diversas pesquisas sobre algoritmos de inteligência artificial, de similaridade e sobre abordagens de filtragem de dados. Ademais, realizamos uma análise profunda de como dados, que, no caso deste trabalho, foram retirados de um banco de dados real e de uma empresa pertinente, podem ser manipulados e modificados para poderem ser aplicados em um algoritmo matemático.

Com isto em vista, o trabalho conseguiu realizar o seu objetivo. Após analisar diferentes tipos de modelos, concluímos que uma abordagem de filtragem de dados baseada

em características dos produtos, somada a um pré-processamento denominado aqui de *flags,* juntamente com um algoritmo de *clustering* como o *k-means,* representa a melhor maneira de criar um sistema de recomendação. Além disso, como nosso trabalho estabeleceu um processo de análise detalhado, a partir de um passo a passo criterioso, bem como a apresentação de diferentes abordagens, de maneira que outros pesquisadores e futuros cientistas de dados podem criar seus próprios modelos, utilizando este trabalho como referência.

Por fim, é importante realçar a importância do tema de sistemas de recomendação de plataformas de *streaming*, especialmente considerando que cada vez mais diversos diferentes produtos estão migrando para esse modelo de negócio. Empresas multinacionais e bilionárias, como a Netflix, baseiam seu sucesso em *streaming* , tornando essencial o uso de um sistema de recomendação eficiente (Anandhan *et al*., 2018).Consequentemente, é importante que futuros pesquisadores continuem a trabalhar e se aprofundar sobre este tema, para que cada vez mais estes serviços ofereçam melhor performance aos seus usuários. Esperamos que esse trabalho possa contribuir de alguma forma para isso.

6. REFERÊNCIAS

ANANDHAN, Anitha et al. Social Media Recommender Systems: Review and Open Research Issues. **IEEE Access**, vol. 6, p. 15608-15628, 2018. Disponível em: https://doi.org/10.1109/access.2018.2810062. Acesso em 27/11/2023.

BEEL, J. et al. Research-paper recommender systems: a literature survey. **International Journal on Digital Libraries**, v. 17, n. 4, p. 305–338, 26 jul. 2015.

CAZELLA, S. C.; NUNES, M. A. S. N.; REATEGUI, E. A Ciência da Opinião: Estado da arte em Sistemas de Recomendação. **JAI Jorn. Atualização em Informática da SBC**. Rio Janeiro, RJ PUC Rio, 161-216. 2010

CUI, M. Introduction to the K-Means Clustering Algorithm Based on the Elbow Method. **Shandong University of Finances and Economics**. Jinan, Shandong, v. 1, 2020.

DAUD, A.; SHAIKH, M. A.; RAJPAR, A. H. Scientific Reference Mining using Semantic Information through Topic Modeling. **Research Journal of Engineering & Technology**, Jamshoro (PK), v. 28, n.2, p. 253—262, jan. /mar. 2009. Disponível em /https://www.academia.edu/32093369. Acesso em 15/06/2023.

DAVIS, N. 2019. **Steam Store Games (Clean dataset**) *in* **Kaggle.** Disponível em: <https://www.kaggle.com/datasets/nikdavis/steam-store-games>. Acesso em: 14/01/2023

FORBES, 2023. **Daniel Elk**. Disponível em: <https://www.forbes.com/profile/danielek/?sh=5867e2e446ab>. Acesso em: 13/06/2023

GOLDBERG, D. et al. Using collaborative filtering to weave an information tapestry. **Communications of the ACM**, v. 35, n. 12, p. 61–70, 1 dez. 1992.

KIM, S. W.; GIL, J. M. Research paper classification systems based on TF-IDF and LDA schemes. **Hum. Cent. Comput. Inf. Sci,** v. 9, n. 30, aug. 2019.

MACQUEEN, J. Some methods for classification and analysis of multivariate observations. **Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability**, v. 5.1, p. 281-297, jan. 1967.

MITRAUD, G. **Recomendacao-Steam.** Repositório do GitHub, 2023. Disponível em: <https://github.com/Gustavo-V-M/Recomendacao-Steam>. Acesso em: 15/06/2023.

MÜLLER, A. C.; GUIDO, S. **Introduction to machine learning with Python : a guide for data scientists**. Beijing: O'reilly, 2016.

OXFORD Dictionaries Online**. The OEC: Facts about the language**. Disponível em: <https://web.archive.org/web/20111226085859/http://oxforddictionaries.com/words/the-oecfacts-about-the-language>. Acesso em: 15/06/2023.

RAHUTOMO, F.; KITASUKA, T.; ARITSUGI, M. Semantic Cosine Similarity**. The 7th International Student Conference on Advanced Science and Technology ICAST 2012**, oct. 2012.

RESNICK, P.; VARIAN, H. R. Recommender systems. **Communications of the ACM**, v. 40, n. 3, p. 56–58, 1 mar. 1997.

SAYER, M.; WILDE, T. 2022. **The 15-year evolution of Steam**. Disponível em: <https://www.pcgamer.com/steam-versions/>. Acesso em: 16/06/2023.

SHAH, D (2022). **Mean Average Precision (mAP) Explained: Everything You Need to Know**. Disponivel em: <https://www.v7labs.com/blog/mean-average-precision>. Acesso em: 17/06/2023.

SINGH, A.; YADAV, A.; RANA, A. K-means with Three different Distance Metrics. **International Journal of Computer Applications**, v. 67, n. 10, p. 13–17, 18 abr. 2013.

STEAM, 2023**. Steam Charts**. Disponível em: <https://store.steampowered.com/charts/>. Acesso em: 17/06/2023

UMARGONO, E.; SUSENO, J. E.; GUNAWAN, S. K. V. K-Means Clustering Optimization Using the Elbow Method and Early Centroid Determination Based on Mean and Median Formula. **Proceedings of the 2nd International Seminar on Science and Technology (ISSTEC 2019),** p. 121-129, oct. 2020.

WISE, J. 2023**. How Many Games are on Steam in 2022? - EarthWeb**. Disponível em: <https://earthweb.com/how-many-games-are-on-steam/>. Acesso em: 15/06/2023.

Contatos: gus_vila_mitra@hotmail.com e leandro.fernandes@mackenzie.br