Descoberta de Subgrupos em Jogos da Steam

Etelvina Oliveira, Henrique Furst Scheid, Igor Eduardo Braga, Indra Matsiendra Ribeiro

[1]Universidade Federal de Minas Gerais (UFMG)

[2]Departamento de Ciência da Computação (DCC-UFMG)

Abstract. *This study applies descriptive learning concepts using the SD Map and Beam Search algorithms to discover subgroups in the 2024 Steam games dataset. Implementations of these algorithms were derived from the subgroups and pysubgroup libraries, respectively. Due to the similar rules found by both algorithms we presented dataset analyses together as a description of which attributes affect the most in characteristics such as price, metacritic score, etc. Additionally, we observed how the data distribution had a significant impact on discovered subgroups.*

Resumo. *Este artigo aplica conceitos de aprendizagem descritiva usando os algoritmos de SD Map e Beam Search para descoberta subgrupos no dataset de jogos Steam de 2024. As implementações desses algoritmos provêm das bibliotecas subgrups e pysubgroup, respectivamente. Devido às regras semelhantes encontradas por ambos os algoritmos, apresentamos as análises em conjunto, como uma descrição de quais atributos do dataset afetam características como preço, pontuação no Metacritic, etc. Além disso, observamos como a distribuição dos dados teve um impacto significativo nos subgrupos descobertos.*

1. Introdução

O presente trabalho foi desenvolvido com o intuito de colocar em prática os conceitos aprendidos na disciplina de Aprendizado Descritivo. Para tanto, utilizamos os algoritmos *SD Map* e Beam Search vistos em sala para realizar a descoberta de subgrupos no dataset de jogos da Steam 2024. Vale ressaltar que as implementações dos algoritmos advém das bibliotecas subgroups e pysubgroup, respectivamente [Lemmerich and Becker 2018].

Analisamos os jogos com base em 4 características principais: avaliação no Metacritc, preço, idade mínima requerida para jogar e sistemas operacionais para os quais o jogo está disponível. Desse modo, verificamos que, no geral, o Beam Search se mostrou tão eficaz quanto a busca exaustiva feita pelo *SD Map*, com ambos trazendo subgrupos semelhantes. Além disso, pudemos observar, na prática, o quanto a qualidade dos dados do dataset e como a escolha dos valores usados para categorizar uma coluna impactam na qualidade dos subgrupos descobertos.

2. Dataset

O projeto foi desenvolvido sobre a base de dados de jogos da Steam, criada pelo repositório Steam Games Scraper que obtém os dados a partir da Steam API e Steam Spy. Esse dataset contém jogos lançados desde 1997 a maio de 2024 e está disponível em https://www.kaggle.com/datasets/artermiloff/steam-games-dataset.

A filtragem inicial realizada no dataset envolveu selecionar colunas de interesse para a mineração de subgrupos, já que o grande número de atributos textuais dificultava o carregamento do dataset e seu processamento. Após análise, optamos por manter as seguintes colunas:

- **dlc_count**: conteúdo adicional que pode ser comprado dentro do jogo.
- **screenshots**: total de prints de tela do jogo tirados por jogadores.
- **achievements**: troféus que ficam visíveis na loja.
- **peak_ccu**: peak concurrent user count, maior número de jogadores jogando online simultaneamente obtido pelo jogo.
- **pct_pos_total**: porcentagem de avaliações positivas no total.
- **name**: nome completo do jogo.
- **price**: Preço do jogo em dólares americanos.
- **windows**: Indica se o jogo é compatível com o sistema operacional Windows .
- **mac**: Indica se o jogo é compatível com o sistema operacional MacOS (False para não compatível, True para compatível).
- **linux**: Indica se o jogo é compatível com o sistema operacional Linux (False para não compatível, True para compatível).
- **metacritic_score**: Pontuação do jogo no Metacritic. (varia de 0 a 1 e representa uma porcentagem)
- **supported_languages**: Lista de idiomas disponíveis para o jogo.
- **categories**: Categorias nas quais o jogo se enquadra (por exemplo, multijogador, single-player).
- **genres**: Gêneros do jogo (por exemplo, ação, aventura).
- **user_score**: Pontuação média dada pelos usuários.
- **estimated_owners**: Estimativa do número de proprietários do jogo.
- **average_playtime_forever**: Tempo médio de jogo (em horas) que os jogadores passaram no jogo.
- **median_playtime_forever**: Tempo mediano de jogo (em horas) que os jogadores passaram no jogo.
- **tags**: Tags associadas ao jogo (por exemplo, FPS, Shooter).
- **pct_pos_total**: Percentual de avaliações positivas do jogo.
- **num_reviews_total**: Número total de avaliações recebidas pelo jogo.

Em seguida foram aplicados diferentes tratamentos às colunas, de acordo com o tipo de dados de cada uma. Desse modo, colunas cuja quantidade poderia ser indicativo de importância do jogo, como por exemplo idiomas para os quais o jogo está disponível, foram convertidas de listas textuais para atributos numéricos indicando a quantidade de línguas para as quais há tradução.

Outras colunas, no entanto, devem ser melhoradas para que tragam alguma informação contemplável pelos algoritmos de aprendizado descritivo [Meeng and Knobbe 2021]. Em alguns casos é útil determinar se um valor está presente, como "support_url" e "support_email" simplesmente salvando a coluna como booleana.

Nessa perspectiva, foram criadas funções para se tratarem as colunas com tipos de dados mais complexos:

- **get_numeric_columns(data)**: recebe um DataFrame como parâmetro e seleciona especificamente as colunas 'required_age', 'price', 'average_playtime_forever', 'median_playtime_forever', 'num_reviews_total', 'peak_ccu', 'pct_pos_total', 'dlc_count' e 'achievements'. Em seguida, ela converte os valores dessas colunas para o tipo float, retornando um novo DataFrame contendo apenas essas colunas com os dados convertidos. Essa função é usada para preparar um conjunto de dados para análises ou modelos que requerem colunas numéricas com um tipo de dado específico.
- **treat_json(x)**: processa uma string JSON para corrigir formatações e converte em um objeto Python. Ela substitui "'s" por "_s", "'em" por "_em", "'Em" por "_Em", e todas as aspas simples por aspas duplas. Essas modificações garantem que a string esteja no formato adequado para ser interpretada como JSON. Em seguida, a função utiliza json.loads para converter a string tratada em um objeto Python e o retorna. Essa função foi utilizada para padronizar e carregar strings JSON com formatações irregulares.
- **bool_to_num(x)**: recebe uma série ou coluna de um DataFrame contendo valores booleanos e converte esses valores para numéricos, aplicando float e depois convertendo para o tipo np.float32. Essa função é usada para transformar variáveis booleanas em um formato adequado para análise numérica.
- **exists_num(x)**: recebe uma série ou coluna de um DataFrame e retorna uma série de valores booleanos indicando se os valores são maiores que zero. Essa função é utilizada para detectar se há dados numéricos na coluna.
- **exists_text(x)**: recebe uma série ou coluna de um DataFrame e verifica se os valores não estão vazios. Retorna uma série onde valores não vazios são convertidos para 1.0 (float) e valores vazios para 0.0, com todos os valores convertidos para o tipo np.float32. Essa função é utilizada para transformar variáveis textuais em um formato numérico, indicando a presença ou ausência de texto.
- **flatten_sum(x)**: recebe uma string JSON, aplica a função treat_json para convertê-la em um objeto Python, e calcula a soma dos valores. Em caso de erro, retorna a soma acumulada até o momento, convertida para o tipo np.float32. Essa função é usada para somar os valores de um JSON convertido, tratando exceções.
- **_count_items(x)**: recebe uma string e conta o número de vírgulas nela. Se houver vírgulas, retorna o número de vírgulas mais um; caso contrário, retorna zero. Essa função é usada internamente para contar itens em uma string separada por vírgulas.
- **count_items(x)**: recebe uma série ou coluna de um DataFrame e aplica a função _count_items a cada valor, retornando uma série com o resultado das contagens, convertida para o tipo np.float32. Essa função é usada para contar o número de itens em strings separadas por vírgulas dentro de um DataFrame.
- **remove_spaces_from_list(string_list)**: recebe uma lista de strings e remove os espaços de cada string na lista, retornando uma nova lista sem espaços. Essa função é utilizada para limpar espaços em branco de elementos de uma lista de strings.
- **get_non_numeric_columns(data)**: recebe um DataFrame e realiza uma série de transformações nas colunas não numéricas. Primeiro, faz uma cópia do DataFrame original. Em seguida, aplica a função exists_num na coluna *metacritic_score* para verificar a existência de valores numéricos. Converte a coluna

supported_languages para o número de itens utilizando a função count_items, e a coluna *tags* para a soma dos valores dentro de cada item JSON usando a função flatten_sum. As colunas *categories* e *genres* também são convertidas para o número de itens utilizando a função *count_items*. Por fim, retorna um novo DataFrame contendo apenas as colunas transformadas: *metacritic_score*, *windows*, *mac*, *linux*, *supported_languages*, *tags*, *categories* e *genres*. Esta função é utilizada para adaptar colunas não numéricas de um DataFrame para um formato mais adequado para análise.

3. Modelo

Utilizamos o *SD Map* para fazer a descoberta de subgrupos devido à sua eficiência e robustez na identificação de padrões interessantes dentro dos dados, por ser um método exaustivo [Atzmueller and Puppe 2006]. O *SD Map* (Subgroup Discovery Map) é uma abordagem que permite encontrar subgrupos de dados que possuem características específicas, diferenciando-os do restante do conjunto de dados com base em medidas de qualidade predefinidas, como a qualidade do subgrupo e a medida de significância estatística.

A biblioteca subgroups em Python facilita essa tarefa ao fornecer ferramentas e métodos para implementar o *SD Map* de maneira intuitiva. Com subgroups, é possível definir facilmente a função de qualidade e as condições para a descoberta de subgrupos, além de realizar a análise de forma eficiente e integrada ao ecossistema Python. Sua interface amigável e bem documentada nos permitiu implementar com sucesso o *SD Map*, testando diferentes contextos e interpretando claramente os resultados. Além disso, utilizamos a biblioteca scikitlearn para realizar a discretização das variaveis, com as funções de **LabelEncoder** e **KBinsDiscretizer**, de modo que o *SD Map* pudesse utilizá-las.

Outrossim, utilizamos o algoritmo de Beam Search [Gamberger and Lavrac 2002] implementado pela biblioteca pysubgroup visando comparar os subgrupos gerados por esse algoritmo heurístico com os subgrupos gerados pela abordagem exaustiva do *SD Map*, tanto em aspectos de qualidade quanto tempo gasto para execução.

4. Métricas

Ao avaliar a coluna *metacritics_score* procurávamos descobrir quais características contribuem para uma melhor avaliação de um jogo nesse quesito. Para tanto, utilizamos a métrica de *lift*, que é um índice estatístico utilizado para definir o grau de interesse de uma regra de associação [Eduardo 2007]. O *lift* mede o aumento na taxa de ocorrência de um evento devido à presença de uma condição específica, através da probabilidade daquele evento ocorrer dada uma condição dividida pela probabilidade do evento ocorrer no geral. Sendo assim, um *lift* igual a 1 indica que a regra de associação não é boa, enquanto valores maiores que 1 são indicativos de uma boa regra.

Além disso, para selecionar os melhores subgrupos tanto no Beam Search quanto no *SD Map* foi utilizada a métrica de WRAcc, que avalia a qualidade de um subgrupo levando em consideração tanto a precisão relativa do subgrupo em relação à população total quanto o tamanho do subgrupo [Vimieiro 2024].

5. Subgrupos descobertos

5.1. Avaliação com base no metacritics_score

O primeiro atributo de interesse avaliado no nosso trabalho foi o *metacritics_score*, por ser um dos principais indicativos da qualidade de um jogo. Primeiramente, avaliamos os jogos que continham alguma avaliação no Metacritc, chegando aos subgrupos na tabela abaixo.

Observamos que há uma grande concentração de jogos com no mínimo 141 avaliações por outros jogadores. É de se esperar que um jogo que tenha mais consumidores seja um jogo mais visível a um site de críticas e, portanto, passível de receber uma pontuação. No entanto, o patamar de 141 reviews é surpreendentemente baixo se considerarmos como os jogos são disponibilizados para todo um mercado mundial simultaneamente.

Adicionalmente, observa-se que o subgrupo com maior *lift* é composto pelos jogos com mais de 141 reviews e preço igual ou superior a 10 dólares (subgrupo de índice 2). Trata-se de um dos subgrupos menos extensos encontrados, que no entanto contém 60% dos jogos avaliados pelo Metacritic. O resultado pode ser interpretado como a preferência pela plataforma de críticas a jogos com algum nível de comercialização, enquanto encontram-se na Steam diversos projetos experimentais com preços simbólicos ou nulos.

A principal observação é que a conjunção "AND windows=True" não altera em nada os resultados de cada subgrupo, pois trata-se do sistema operacional dominante dentro da indústria de jogos, e virtualmente não há lançamento que exclua essa plataforma.

Em seguida, avaliamos os subgrupos descobertos para os jogos com pontuação maior que 0.8 no *metacritcs_score*, que foram considerados como jogos bem avaliados. Nesse sentido, foi possível perceber que os atributos que mais impactaram para uma boa avaliação no *metacritcs_score* foram o pct_pos_total maior ou igual a 92, um *num_total_reviews* maior ou igual a 7625 e um *peak_ccu* maior ou igual a 82, indicando que jogos bem avaliados recebem mais reviews, têm uma alta porcentagem de avaliaçõe positivas e consequentemente são jogados por muitas pessoas (figura 1). Ao passo que jogos com baixa pontuação, menor que 0.8 no *metacritics_score*, tiveram subgrupos com baixo peak_ccu, entre 0 e 2, e baixo *pct_pos_total*, menor ou igual a 69 (figura 2).

subgroup	relative_size_sg	coverage_sg	lift
pct_pos_total>=92.0	0.22106598984771575	0.48981670061099797	2.2157035595951
pct_pos_total>=92.0 AND windows==True	0.22106598984771575	0.48981670061099797	2.2157035595951
pct_pos_total>=92.0 AND tags==0.0	0.22106598984771575	0.48981670061099797	2.2157035595951
pct_pos_total>=92.0 AND tags==0.0 AND windows==True	0.22106598984771575	0.48981670061099797	2.2157035595951
pct_pos_total>=92.0 AND required_age: [0.0:1.0[	0.19365482233502537	0.4164969450101833	2.150718169515232
pct_pos_total>=92.0 AND required_age: [0.0:1.0[AND windows==True	0.19365482233502537	0.4164969450101833	2.150718169515232
pct_pos_total>=92.0 AND required_age: [0.0:1.0[AND tags==0.0	0.19365482233502537	0.4164969450101833	2.150718169515232
pct_pos_total>=92.0 AND required_age: [0.0:1.0[AND tags==0.0 AND windows==True	0.19365482233502537	0.4164969450101833	2.150718169515232
num_reviews_total>=7625.0	0.2	0.4164969450101833	2.0824847250509166
num_reviews_total>=7625.0 AND windows==True	0.2	0.4164969450101833	2.0824847250509166
num_reviews_total>=7625.0 AND tags==0.0	0.2	0.4164969450101833	2.0824847250509166
num_reviews_total>=7625.0 AND tags==0.0 AND windows==True	0.2	0.4164969450101833	2.0824847250509166
peak_ccu>=82.0	0.20025380710659899	0.4134419551934827	2.0645897382285447
peak_ccu>=82.0 AND windows==True	0.20025380710659899	0.4134419551934827	2.0645897382285447
peak_ccu>=82.0 AND tags==0.0	0.20025380710659899	0.4134419551934827	2.0645897382285447
peak_ccu>=82.0 AND tags==0.0 AND windows==True	0.20025380710659899	0.4134419551934827	2.0645897382285447
num_reviews_total>=7625.0 AND peak_ccu>=82.0	0.1515228426395939	0.34826883910386963	2.2984576651076156
num_reviews_total>=7625.0 AND peak_ccu>=82.0 AND windows==True	0.1515228426395939	0.34826883910386963	2.2984576651076156
num_reviews_total>=7625.0 AND peak_ccu>=82.0 AND tags==0.0	0.1515228426395939	0.34826883910386963	2.2984576651076156
num_reviews_total>=7625.0 AND peak_ccu>=82.0 AND tags==0.0 AND windows==True	0.1515228426395939	0.34826883910386963	2.2984576651076156

Figure 1. Subgrupos de interesse encontrados para metacritics_score maior ou igual a 80

subgroup	relative_size_sg	coverage_sg	lift
peak_ccu: [0.0:2.0[	0.35076142131979693	0.4148073022312373	1.1825910063611251
peak_ccu: [0.0:2.0[AND windows==True	0.35076142131979693	0.4148073022312373	1.1825910063611251
peak_ccu: [0.0:2.0[AND tags==0.0	0.35076142131979693	0.4148073022312373	1.1825910063611251
peak_ccu: [0.0:2.0[AND tags==0.0 AND windows==True	0.35076142131979693	0.4148073022312373	1.1825910063611251
peak_ccu: [0.0:2.0[AND required_age: [0.0:1.0[	0.32944162436548224	0.3894523326572008	1.1821588525958175
peak_ccu: [0.0:2.0[AND required_age: [0.0:1.0[AND windows==True	0.32944162436548224	0.3894523326572008	1.1821588525958175
peak_ccu: [0.0:2.0[AND required_age: [0.0:1.0[AND tags==0.0	0.32944162436548224	0.3894523326572008	1.1821588525958175
peak_ccu: [0.0:2.0[AND required_age: [0.0:1.0[AND tags==0.0 AND windows==True	0.32944162436548224	0.3894523326572008	1.1821588525958175
pct_pos_total<69.0	0.1862944162436548	0.2376605814739689	1.2757257370673534
pct_pos_total<69.0 AND windows==True	0.1862944162436548	0.2376605814739689	1.2757257370673534
pct_pos_total<69.0 AND tags==0.0	0.1862944162436548	0.2376605814739689	1.2757257370673534
pct_pos_total<69.0 AND tags==0.0 AND windows==True	0.1862944162436548	0.2376605814739689	1.2757257370673534
pct_pos_total<69.0 AND required_age: [0.0:1.0[	0.16218274111675127	0.20655848546315078	1.2736157006648108
pct_pos_total<69.0 AND required_age: [0.0:1.0[AND windows==True	0.16218274111675127	0.20655848546315078	1.2736157006648108
pct_pos_total<69.0 AND required_age: [0.0:1.0[AND tags==0.0	0.16218274111675127	0.20655848546315078	1.2736157006648108
pct_pos_total<69.0 AND required_age: [0.0:1.0[AND tags==0.0 AND windows==True	0.16218274111675127	0.20655848546315078	1.2736157006648108
linux==False AND peak_ccu: [0.0:2.0[	0.2530456852791878	0.2964841108857336	1.1716623840419162
linux==False AND peak_ccu: [0.0:2.0[AND windows==True	0.2530456852791878	0.2964841108857336	1.1716623840419162
linux==False AND peak_ccu: [0.0:2.0[AND tags==0.0	0.2530456852791878	0.2964841108857336	1.1716623840419162
linux==False AND peak_ccu: [0.0:2.0[AND tags==0.0 AND windows==True	0.2530456852791878	0.2964841108857336	1.1716623840419162

Figure 2. Subgrupos de interesse encontrados para metacritics_score menor que 80

5.2. Avaliação com base na indicação etária

Decidimos analisar os subgrupos de jogos com idade recomendada menor que 16 anos, uma característica principal de jogos livres para todo público. Observamos que os grupos com maior qualidade (*quality*) e *lift* todos contém *median_playtime_forever*, *average_playtime_forever* e *peak_ccu*, entre 0 e 1. A partir desse resultado, cujos subgrupos cobriam em torno de 70% dos jogos dessa indicação etária, estima-se que a grande maioria sejam programas não comercializados (como ferramentas ou versões de teste) ou fracassos de vendas. Subgrupos delimitados pelo intervalo [0, 20) para a variável *metacritic_score* também foram recorrentes. Observou-se que em sua maioria os jogos desse subgrupo possuíam avaliação 0, e poucas exceções ampliaram o intervalo para 20. Entende-se que 0 seja o valor padrão para jogos não-avaliados pela plataforma *Metacritic*, a razão para a qual sendo a mesma não-comercialidade discorrida acima.

5.3. Avaliação com base no preço

Ao avaliarmos subgrupos gerados tanto pelo beam search quanto pelo *SD Map*, optamos por dividi-los em duas categorias: jogos com preço acima ou abaixo da média observada. Porém, devido ao grande número de softwares gratuitos catalogados na Steam, a média foi rebaixada para USD$7,49, um valor não-correspondente com o preço comum de um jogo comercial. As análises para jogos precificados acima da média não tivevam boa qualidade. Ao se elevar o limite de preço para USD $60,00 (um valor mais comumente tido como um jogo "caro") foram encontrados subgrupos de maior interesse, mas ainda limitados por colunas com muitos valores-padrão. Uma filtragem de valores-padrão sobre as colunas *num_reviews_total*, *peak_ccu*, *average_playtime_forever* e *metacritic_score* retornou valores surpreendentes. Todas as descrições encontradas cobrem todos os jogos "caros". Descrições incluindo tempo médio jogado acima de 1062h e a infrequência de suporte a *mac* e *linux* (figura 3).

subgroup	relative_size_sg	coverage_sg	lift
dlc_count>=3.0 AND linux==False AND mac==False	0.13884297520661157	0.62	4.46547619047619
dlc_count>=3.0 AND linux==False AND mac==False AND windows==True	0.13884297520661157	0.62	4.46547619047619
dlc_count>=3.0 AND linux==False AND mac==False AND tags==0.0	0.13884297520661157	0.62	4.46547619047619
dlc_count>=3.0 AND linux==False AND mac==False AND tags==0.0 AND windows==True	0.13884297520661157	0.62	4.46547619047619
dlc_count>=3.0 AND mac==False	0.14269972451790633	0.62	4.344787644787645
dlc_count>=3.0 AND mac==False AND windows==True	0.14269972451790633	0.62	4.344787644787645
dlc_count>=3.0 AND mac==False AND tags==0.0	0.14269972451790633	0.62	4.344787644787645
dlc_count>=3.0 AND mac==False AND tags==0.0 AND windows==True	0.14269972451790633	0.62	4.344787644787645
average_playtime_forever>=1062.0 AND linux==False AND mac==False	0.1305785123966942	0.6	4.59493670886076
average_playtime_forever>=1062.0 AND linux==False AND mac==False AND windows==True	0.1305785123966942	0.6	4.59493670886076
average_playtime_forever>=1062.0 AND linux==False AND mac==False AND tags==0.0	0.1305785123966942	0.6	4.59493670886076
average_playtime_forever>=1062.0 AND linux==False AND mac==False AND tags==0.0 AND windows==True	0.1305785123966942	0.6	4.59493670886076
average_playtime_forever>=1062.0 AND linux==False	0.15537190082644628	0.62	3.9904255319148936
average_playtime_forever>=1062.0 AND linux==False AND windows==True	0.15537190082644628	0.62	3.9904255319148936
average_playtime_forever>=1062.0 AND linux==False AND tags==0.0	0.15537190082644628	0.62	3.9904255319148936
average_playtime_forever>=1062.0 AND linux==False AND tags==0.0 AND windows==True	0.15537190082644628	0.62	3.9904255319148936
average_playtime_forever>=1062.0 AND mac==False	0.13663911845730028	0.6	4.391129032258065
average_playtime_forever>=1062.0 AND mac==False AND windows==True	0.13663911845730028	0.6	4.391129032258065
average_playtime_forever>=1062.0 AND mac==False AND tags==0.0	0.13663911845730028	0.6	4.391129032258065
average_playtime_forever>=1062.0 AND mac==False AND tags==0.0 AND windows==True	0.13663911845730028	0.6	4.391129032258065

Figure 3. Subgrupos de interesse encontrados para jogos com preço acima de 59 dólares

Nos principais subgrupos descobertos para jogos de preço abaixo da média (figura 4) encontram-se duas características de suma importância: a baixa pontuação no *metacritic_score* entre 0 e 0.2, e um baixo valor de *peak_ccu*, entre 0 e 1, forte indício de nenhuma ou baixa comercialização do jogo. Podem tratar-se de demos gratuitas, jogos com fracasso comercial ou versões de teste não comercializadas. Entretanto, existem outras partes das regras que são consequências das características do dataset, por exemplo, acreditamos que o atributo windows=true seja consequência do fato de mais de 90% da base de dados ter essa característica, enquanto mac e linux apenas 20% e 14% respectivamente.

subgroup	relative_size_sg	coverage_sg	lift
metacritic_score: [0:20[AND peak_ccu: [0.0:1.0[AND required_age: [0.0:1.0[AND windows==True	0.7586136814671353	0.8492292085575059	1.1194488437317964
metacritic_score: [0:20[AND peak_ccu: [0.0:1.0[AND required_age: [0.0:1.0[AND tags==0.0 AND windows==True	0.7586136814671353	0.8492292085575059	1.1194488437317964
metacritic_score: [0:20[AND peak_ccu: [0.0:1.0[AND required_age: [0.0:1.0[	0.7589364703631973	0.8495009329203123	1.1193307557268586
metacritic_score: [0:20[AND peak_ccu: [0.0:1.0[AND required_age: [0.0:1.0[AND tags==0.0	0.7589364703631973	0.8495009329203123	1.1193307557268586
metacritic_score: [0:20[AND peak_ccu: [0.0:1.0[AND windows==True	0.7646151639050284	0.8548086154737967	1.1179592765439468

Figure 4. Subgrupos de interesse encontrados para preço menor que a média de USD $7,49

5.4. Avaliação com base no sistema operacional

Para estudar os subgrupos nos sistemas operacionais, decidimos analisar individualmente cada um: Linux, Windows e Mac.

Começamos com o sistema Mac (figura 5), onde conseguimos obter informações relevantes. Primeiro, identificamos subgrupos com $pct\_pos\_total \geq 88$, indicando que os jogos disponíveis para Mac geralmente são muito bem avaliados. Outra característica interessante encontrada foi $supported\_languages \geq 6.0$, indicando que muitos jogos para Mac suportam diversos idiomas. Além disso, notamos que alguns subgrupos apresentaram $categories \geq 6.0$, revelando que o Mac possui muitos jogos com várias categorias. Isso sugere uma grande diversidade de conteúdo, atendendo a diferentes preferências e interesses dos jogadores. Essas características sugerem que os jogos disponíveis para Mac geralmente têm um alto padrão de qualidade, pois este sistema operacional é conhecido por suas exigências rigorosas em relação à disponibilidade de aplicativos e jogos. Além disso, esse fato pode indicar que jogos desse sistema operacional geralmente recebem um maior investimento financeiro. Isso pode indicar que desenvolvedores e empresas estão dispostos a investir mais em jogos para Mac, resultando em produtos mais bem construídos e com melhores avaliações.

No Linux (figura 6), observamos características semelhantes às do Mac. Subgrupos com $categories \geq 6.0$, $pct\_pos\_total \geq 88.0$ e $achievements \geq 24.0$ foram identificados. Isso indica que, assim como no Mac, os jogos para Linux tendem a ter alta qualidade percebida, suporte a diversas categorias e um número significativo de conquistas. Isso

No Windows, observamos características um pouco diferentes. Identificamos subgrupos com *categories* variando de 4 a 6, sugerindo uma diversidade moderada de conteúdo (figura 7). Isso indica que, ao contrário dos outros sistemas operacionais, não há tanto apelo econômico na disponibilidade desses jogos, já que o Windows é conhecido por ser um sistema mais aberto e acessível em relação à oferta de jogos. Dessa forma, diferentes tipos de jogos, até mesmo os mais simples, podem ser disponibilizados mais facilmente para os usuários da Steam.

subgroup	relative_size_sg	coverage_sg	lift
num_reviews_total>=141.0	0.20005738469263323	0.3011170180992772	1.5051532267199799
num_reviews_total>=141.0 AND tags==0.0	0.20005738469263323	0.3011170180992772	1.5051532267199799
pct_pos_total>=88.0	0.20681204122133753	0.2956215279851861	1.4294212572893736
pct_pos_total>=88.0 AND tags==0.0	0.20681204122133753	0.2956215279851861	1.4294212572893736
categories>=6.0	0.225390335461349	0.31055492503434684	1.3778537770870933
categories>=6.0 AND tags==0.0	0.225390335461349	0.31055492503434684	1.3778537770870933
achievements>=24.0	0.2080195107954953	0.2881548294606057	1.3852298198311392
achievements>=24.0 AND tags==0.0	0.2080195107954953	0.2881548294606057	1.3852298198311392
supported_languages>=6.0	0.20237668268656003	0.27794038587897973	1.3733814695908046
supported_languages>=6.0 AND tags==0.0	0.20237668268656003	0.27794038587897973	1.3733814695908046
median_playtime_forever>=3.0	0.1125696387155393	0.18690639746729587	1.6603624174330318
median_playtime_forever>=3.0 AND tags==0.0	0.1125696387155393	0.18690639746729587	1.6603624174330318
average_playtime_forever>=3.0	0.11250986299404633	0.18678693029090257	1.66018271927668
average_playtime_forever>=3.0 AND tags==0.0	0.11250986299404633	0.18678693029090257	1.66018271927668
average_playtime_forever>=3.0 AND median_playtime_forever>=3.0	0.11243813212825479	0.1866674631145093	1.6601793322356453
average_playtime_forever>=3.0 AND median_playtime_forever>=3.0 AND tags==0.0	0.11243813212825479	0.1866674631145093	1.6601793322356453
metacritic_score: [0:20[AND pct_pos_total>=88.0	0.1890466967936303	0.2545845528940924	1.3466754892417157
metacritic_score: [0:20[AND pct_pos_total>=88.0 AND tags==0.0	0.1890466967936303	0.2545845528940924	1.3466754892417157
metacritic_score: [0:20[AND num_reviews_total>=141.0	0.1608445113932525	0.2242398900901977	1.3941407645670194
metacritic_score: [0:20[AND num_reviews_total>=141.0 AND tags==0.0	0.1608445113932525	0.2242398900901977	1.3941407645670194

Figure 5. Subgrupos de interesse encontrados para jogos disponíveis para Mac

subgroup	relative_size_sg	coverage_sg	lift
categories>=6.0	0.225390335461349	0.3465237001106289	1.5374381488067503
categories>=6.0 AND tags==0.0	0.225390335461349	0.3465237001106289	1.5374381488067503
achievements>=24.0	0.2080195107954953	0.30422942728278446	1.4625042916376891
achievements>=24.0 AND tags==0.0	0.2080195107954953	0.30422942728278446	1.4625042916376891
pct_pos_total>=88.0	0.20681204122133753	0.30057016424134114	1.4533494397439866
pct_pos_total>=88.0 AND tags==0.0	0.20681204122133753	0.30057016424134114	1.4533494397439866
num_reviews_total>=141.0	0.20005738469263323	0.28797549144753637	1.4394644411151325
num_reviews_total>=141.0 AND tags==0.0	0.20005738469263323	0.28797549144753637	1.4394644411151325
categories>=6.0 AND metacritic_score: [0:20[	0.19832388876933746	0.2827844438771168	1.4258718182256505
categories>=6.0 AND metacritic_score: [0:20[AND tags==0.0	0.19832388876933746	0.2827844438771168	1.4258718182256505
achievements: [10.0:24.0[	0.20720656098319107	0.28899668113352056	1.3947274630795325
achievements: [10.0:24.0[AND tags==0.0	0.20720656098319107	0.28899668113352056	1.3947274630795325
dlc_count: [1.0:2.0[	0.10195347057838988	0.17462343630329333	1.7127757918650652
dlc_count: [1.0:2.0[AND tags==0.0	0.10195347057838988	0.17462343630329333	1.7127757918650652
categories>=6.0 AND median_playtime_forever: [0.0:1.0[	0.1730746240107118	0.24534082205769722	1.4175435795978546
categories>=6.0 AND median_playtime_forever: [0.0:1.0[AND tags==0.0	0.1730746240107118	0.24534082205769722	1.4175435795978546
average_playtime_forever: [0.0:1.0[AND categories>=6.0	0.1730746240107118	0.24534082205769722	1.4175435795978546
average_playtime_forever: [0.0:1.0[AND categories>=6.0 AND tags==0.0	0.1730746240107118	0.24534082205769722	1.4175435795978546
average_playtime_forever: [0.0:1.0[AND categories>=6.0 AND median_playtime_forever: [0.0:1.0[	0.1730746240107118	0.24534082205769722	1.4175435795978546
average_playtime_forever: [0.0:1.0[AND categories>=6.0 AND median_playtime_forever: [0.0:1.0[AND tags==0.0	0.1730746240107118	0.24534082205769722	1.4175435795978546

Figure 6. Subgrupos de interesse encontrados para jogos disponíveis para Linux

subgroup	relative_size_sg	coverage_sg	lift
categories: [4.0:6.0[	0.27359347727327O7	0.2736916379640260	1.0003587830080367
categories: [4.0:6.0[AND tags==0.0	0.27359347727327O7	0.2736916379640260	1.0003587830080367
categories: [4.0:6.0[AND metacritic_score: [0:20[	0.26185352557205366	0.26194747416762343	1.0003587830080367
categories: [4.0:6.0[AND metacritic_score: [0:20[AND tags==0.0	0.26185352557205366	0.26194747416762343	1.0003587830080367
categories: [4.0:6.0[AND median_playtime_forever: [0.0:1.0[	0.23654448509193507	0.23662935323383086	1.0003587830080367
categories: [4.0:6.0[AND median_playtime_forever: [0.0:1.0[AND tags==0.0	0.23654448509193507	0.23662935323383086	1.0003587830080367
average_playtime_forever: [0.0:1.0[AND categories: [4.0:6.0[	0.23654448509193507	0.23662935323383086	1.0003587830080367
average_playtime_forever: [0.0:1.0[AND categories: [4.0:6.0[AND tags==0.0	0.23654448509193507	0.23662935323383086	1.0003587830080367
average_playtime_forever: [0.0:1.0[AND categories: [4.0:6.0[AND median_playtime_forever: [0.0:1.0[	0.23654448509193507	0.23662935323383086	1.0003587830080367
average_playtime_forever: [0.0:1.0[AND categories: [4.0:6.0[AND median_playtime_forever: [0.0:1.0[AND tags==0.0	0.23654448509193507	0.23662935323383086	1.0003587830080367
categories: [4.0:6.0[AND median_playtime_forever: [0.0:1.0[AND metacritic_score: [0:20[	0.23061473351983358	0.23069747416762343	1.0003587830080367
categories: [4.0:6.0[AND median_playtime_forever: [0.0:1.0[AND metacritic_score: [0:20[AND tags==0.0	0.23061473351983358	0.23069747416762343	1.0003587830080367
average_playtime_forever: [0.0:1.0[AND categories: [4.0:6.0[AND metacritic_score: [0:20[	0.23061473351983358	0.23069747416762343	1.0003587830080367
average_playtime_forever: [0.0:1.0[AND categories: [4.0:6.0[AND metacritic_score: [0:20[AND tags==0.0	0.23061473351983358	0.23069747416762343	1.0003587830080367
average_playtime_forever: [0.0:1.0[AND categories: [4.0:6.0[AND median_playtime_forever: [0.0:1.0[AND metacritic_score: [0:20[	0.23061473351983358	0.23069747416762343	1.0003587830080367
average_playtime_forever: [0.0:1.0[AND categories: [4.0:6.0[AND median_playtime_forever: [0.0:1.0[AND metacritic_score: [0:20[AND tags==0.0	0.23061473351983358	0.23069747416762343	1.0003587830080367
genres: [3.0:4.0[AND median_playtime_forever: [0.0:1.0[	0.26056236998780574	0.26064389590508996	1.0003128844632785
genres: [3.0:4.0[AND median_playtime_forever: [0.0:1.0[AND tags==0.0	0.26056236998780574	0.26064389590508996	1.0003128844632785
average_playtime_forever: [0.0:1.0[AND genres: [3.0:4.0[	0.26056236998780574	0.26064389590508996	1.0003128844632785
average_playtime_forever: [0.0:1.0[AND genres: [3.0:4.0[AND tags==0.0	0.26056236998780574	0.26064389590508996	1.0003128844632785

Figure 7. Subgrupos de interesse encontrados para jogos disponíveis para Windows

6. Conclusão

A partir da aplicação dos algoritmos de Beam Search e *SD Map* observamos que na base de dados em questão as regras descobertas por ambos foram semelhantes, mostrando que o Beam Search apesar de heurístico apresenta ótimo desempenho e é computacionalmente menos custoso, conforme estudado em sala. Além disso, foi possível perceber também que, mesmo após a limpeza do dataset, ainda existem "erros" nas regras dos subgrupos que são causados pelas características do próprio dataset, como por exemplo a inserção da regra windows=true em muitos subgrupos, que é uma regra irrelevante, mas que apareceu tantas vezes devido ao fato de mais de 90% do dataset possuir esse atributo como true. Portanto, atributos muito frequentes podem afetar a redundância das regras geradas pelos algoritmos de SD.

Ademais, foi possível perceber que a média e a moda nem sempre são bons critérios para categorizar um atributo, pois a presença de outliers ou uma base com muitos dados de um certo tipo impactam na média. Nesse sentido, a média do preço do dataset da Steam é muito baixa, devido ao grande número de jogos gratuitos ou muito baratos. Portanto, utilizar a média para categorizar um jogo como caro ou barato não foi a melhor opção, em bases desbalanceadas pode ser melhor categorizar um atributo numérico utilizando o range de valores.

Em trabalhos futuros, gostaríamos de estender as análises sobre esse dataset trazendo subgrupos excepcionais, para agregar análises mais inovadoras em relação aos jogos, pois os algoritmos de SD comuns utilizados, apesar de funcionarem corretamente, trouxeram muitos subgrupos que já seriam esperados com base no conhecimento geral do dataset, visto que as regras traziam apenas variáveis que já eram altamente correlacionadas à classe de interesse, por exemplo, era esperado que um jogo bem avaliado na steam também fosse bem avaliado no Metacritic.

References

Atzmueller, M. and Puppe, F. (2006). Sd-map – a fast algorithm for exhaustive subgroup discovery. In Fürnkranz, J., Scheffer, T., and Spiliopoulou, M., editors, *Knowledge Discovery in Databases: PKDD 2006*, pages 6–17, Berlin, Heidelberg. Springer Berlin Heidelberg.

Eduardo (2007). Data mining de regras de associação. Acessso em 30 de jul., 2024.

Gamberger, D. and Lavrac, N. (2002). Expert-guided subgroup discovery: methodology and application. *J. Artif. Int. Res.*, 17(1):501–527.

Lemmerich, F. and Becker, M. (2018). pysubgroup: Easy-to-use subgroup discovery in python. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 658–662.

Meeng, M. and Knobbe, A. (2021). For real: a thorough look at numeric attributes in subgroup discovery. *Data Mining and Knowledge Discovery*, 35:158–212.

Vimieiro, R. (2024). Aula 09 – aprendizado descritivo supervisionado. aula da disciplina de aprendizado descritivo.