Instituto de MATEMÁTICA E ESTATÍSTICA UFRGS

DEPARTAMENTO DE ESTATÍSTICA
E-mail: dest@mat.ufrgs.br

Trabalho de Conclusão de Curso

Análise Preditiva da Popularidade de Jogos Indie

Rafaela Oliveira da Silva

Abril de 2023

Rafaela Oliveira da Silva

Análise Preditiva da Popularidade de Jogos Indie

Trabalho de Conclusão apresentado à comissão de Graduação do Departamento de Estatística da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para obtenção do título de Bacharela em Estatística.

Orientador: Prof. Dr. João Henrique Ferreira Flores

Porto Alegre
Abril de 2023

Rafaela Oliveira da Silva

Análise Preditiva da Popularidade de Jogos Indie

Este Trabalho foi julgado adequado para obtenção dos créditos da disciplina Trabalho de Conclusão de Curso em Estatística e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador:_____
Prof. Dr. João Henrique Ferreira Flores, UFRGS
Universidade Federal do Rio Grande do Sul – Porto Alegre, RS

Banca Examinadora:

Profa. Dra. Márcia Helena Barbian, UFMG
Universidade Federal de Minas Gerais – Belo Horizonte, MG

Prof. Dr. Rodrigo Citton Padilha dos Reis, UFMG
Universidade Federal de Minas Gerais – Belo Horizonte, MG

Porto Alegre
Abril de 2023

Agradecimentos

Gostaria de agradecer a todos aqueles que me incentivaram, apoiaram e me ensinaram nessa segunda graduação em Estatística.

Um agradecimento ao orientador João Henrique Ferreira Flores, por ter acreditado no potencial do projeto de pesquisa e por ter me auxiliado nesse último ano com o trabalho de conclusão.

Um agradecimento também à professora Márcia Helena Barbian e ao professor Rodrigo Citton Padilha dos Reis por aceitarem fazer parte da banca.

Um agradecimento especial às minhas gatas Milka e Tigs, por fazerem minha vida mais feliz. Dedico esse trabalho à Dolly (*in memoriam*), a boxer que fez parte por 15 anos da minha vida.

E por fim, gostaria de agradecer à Carolina Oliveira, minha irmã gêmea e a melhor *sestra* que alguém poderia ter, e ao Nicolas Hahn, meu grande amor e parceiro.

"What do we do when the lights go dark? We grow taller.
Where do we go when the shadows grow? We go further.
How do we wake up from bad dreams? We dream harder.
Why do we stay here when we could just
Leave."

Shady Part of Me

Resumo

Os jogos eletrônicos se tornaram cada vez mais populares na última década e esse crescimento pode ser visto no seu valor de mercado. Além disso, com a grande quantidade de jogos sendo lançados anualmente na plataforma Steam, é necessário que um jogo se destaque em relação a outros. Esse problema é maior para desenvolvedores independentes, que em sua maioria não consegue gerar uma receita bruta vitalícia maior que mil dólares.

Portanto, neste trabalho, temos como objetivo estudar quais características dos jogos *indie* influenciam em diferentes métricas de popularidade: quantidade de avaliações, proporção de avaliações positivas e descrição das avaliações. A principal análise realizada foi o impacto dos diferentes gêneros na modelagem estatística e, devido a isso, foi realizada uma análise de associação e um agrupamento dos diversos gêneros desses jogos. Por fim, os modelos que tiveram melhor desempenho serão aplicados em uma base de jogos *indie* lançados na plataforma Steam em 2023, comparando os resultados com o que se encontra na base de dados.

Palavras-Chave: Jogos, Jogos Eletrônicos, Jogos Independentes, Pontuação de Jogos, Aprendizagem de Máquina.

Abstract

Video games have become very popular in the last decade, and this growth can be seen in their market value. In addition, with a large number of games being released annually on the Steam platform, it is necessary for a game to stand out from the rest. This problem is even more complicated for independent developers, who mostly of them cannot generate lifetime gross revenue greater than one thousand dollars.

Therefore, in this work, we aim to study which characteristics of indie games influence different popularity metrics, the number of reviews, the proportion of positive reviews, and the description of the user score. The principal analysis was the impact of the different genres on the statistical modeling, and for this reason, we made an association analysis and a clustering of the different genres of these games. Finally, the best-performing models will be applied to a database of indie games released on the Steam platform in 2023, comparing the results with the game's data.

Keywords: Games, Video Games, Indie Games, User Score, Machine Learning.

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

Contextualização

A indústria de jogos cresceu muito na última década e tem se tornado cada vez mais popular devido aos avanços tecnológicos, permitindo que mais pessoas tenham acesso a computadores, celulares e consoles. Esse crescimento pode ser visto no seu valor de mercado, que alcançou um valor de 178 bilhões de dólares em 2021 e com uma estimativa de alcançar 268 bilhões de dólares em 2025 (Clemen, 2021).

A Steam, desenvolvida pela Valve Corporation em 2003, é uma das maiores plataformas de distribuição de jogos para computador existentes com um catálogo de mais de 50 mil jogos e tem lançado em torno de 10 mil jogos por ano desde 2020 (SteamDB, 2023b). Além disso, possui uma grande base de jogadores, tendo alcançado mais de 33 milhões de usuários simultâneos em Março de 2023 (SteamDB, 2023a).

Devido a grande quantidade de jogos sendo lançados na plataforma, jogadores podem ter dificuldade em selecionar um jogo a ser comprado ou jogado futuramente, e desenvolvedores independentes podem ter dificuldade nas vendas de seus jogos. Em ambos os casos, é necessário que um jogo se destaque em relação aos outros, atraindo a atenção de novos jogadores. Na Steam, jogos se destacam na página inicial pelo número de vendas, pela quantidade de jogadores simultâneos, por promoções, por boas avaliações feitas pelos usuários e por propagandas pagas.

Um jogo *indie* (independente), conforme Garda e Grabarczyk (2016), pode ser explicado por meio de três tipos de independência: a financeira, a criativa e a de publicação. Ainda, segundo o artigo da VG Insights (2022), desenvolvedores independentes podem ser divididos conforme seus ganhos vitalícios, de forma que apenas 3% desses desenvolvedores ganharam mais de 1 milhão de dólares em receita bruta, sendo considerados as histórias de sucesso.

Trabalhos Relacionados

Trnený (2017) estimou o sucesso de jogos eletrônicos baseado em informações descritivas como gênero, preço, desenvolvedor e requisitos do jogo utilizando modelos de *Machine Learning*. O nível de sucesso de um jogo foi definido como o número médio de jogadores simultâneos nos dois primeiros meses após o lançamento. Neste estudo o autor utilizou os modelos *SVM* e *Random Forest*, que obtiveram os melhores resultados, respectivamente, para jogos com desenvolvedores ou distribuidores com pelo menos dois jogos em seu histórico e para jogos alcançando em média 100 jogadores simultâneos.

De Luisa et al. (2021) utilizaram modelos Bayesianos para predizer a popularidade de jogos na Steam e para entender a influência de preço, tamanho, idiomas, data de lançamento e gênero na sua contagem de jogadores. A popularidade de um jogo foi definida como o número de jogadores jogando um jogo ao longo do tempo e a principal variável de predição foi a contagem mediana de jogadores no segundo mês após o lançamento de um jogo.

Ziyang (2021) teve como foco a predição da popularidade de jogos eletrônicos *indie* (independentes) na plataforma Steam por meio de Regressão Logística e *Random Forest*, utilizando marcadores (*tags*) como principal característica. Os marcadores utilizados foram relacionados com gênero, subgênero, estilo de jogo, visual, recursos, entre outros. Jogos independentes foram considerados jogos com orçamento limitado em termos de processos de desenvolvimento, recursos humanos e publicidade. Além disso, para avaliar a popularidade foi utilizado o número de jogadores que possuem o jogo em vez do número de vendas, uma vez que o número de vendas exato não é revelado ao público geral.

Objetivos

Com o intuito de contribuir com este problema de pesquisa, o objetivo principal deste trabalho é realizar uma análise preditiva da popularidade de jogos *indie* da plataforma Steam. Serão utilizadas 3 diferentes métricas para caracterizar a popularidade de um jogo: (*i*) quantidade de avaliações (*reviews*), (*ii*) proporção de avaliações positivas em relação ao total de avaliações e (*iii*) descrição das avaliações (positivas e não-positivas).

O foco do trabalho é entender quais características influenciam nas métricas de popularidade selecionadas e, principalmente, o entendimento dos grupos de gêneros e seus impactos na modelagem estatística e de *Machine Learning*. Os dados dos jogos foram coletados por meio da Steam Web API e foram utilizados, para análise e modelagem, os jogos *indie* lançados entre 2018 e 2022, e que possuíam pelo menos 5 avaliações feitas por usuários. Ainda, este trabalho irá demonstrar a aplicação de alguns modelos que tiveram melhor desempenho, como *Random Forest* e *Ridge Regression*, em uma base de jogos lançados na plataforma Steam em 2023.

Portanto, esta pesquisa auxiliará jogadores e usuários da Steam a terem mais uma ferramenta para verificar se compensa comprar um determinado jogo a ser lançado, como também ajudará desenvolvedores na escolha de gêneros e características que melhor contribuam para a pontuação de um jogo em desenvolvimento.

2 Jogos

A Steam foi criada pela Valve em 2003 para servir como um canal de distribuição de conteúdo digital para jogos, antes das lojas de aplicativos existirem. Desde então, cresceu e evoluiu para uma plataforma para milhares de criadores e distribuidoras fornecerem conteúdo e estabelecerem relacionamentos diretos com seus clientes. A Comunidade Steam permite que milhões de jogadores façam o mesmo, compartilhando entretenimento, ideias e fazendo amigos (Valve, 2023).

A Steam se considera o melhor destino para jogar, discutir e criar jogos. A plataforma disponibiliza acesso aos jogos instantaneamente, desde jogos AAA até jogos *indie*, com ofertas exclusivas, atualizações automáticas e outras diversas vantagens. Além disso, conta com diversas possibilidades, como ganhar *achievements* nos jogos, ler avaliações feitas por outros jogadores e explorar recomendações personalizadas. Por fim, os desenvolvedores podem utilizar a Steamworks para lançarem seus jogos na plataforma, que é um conjunto de ferramentas e serviços que ajudam os desenvolvedores e as distribuidoras de jogos a aproveitarem ao máximo a distribuição de jogos na Steam (Steam, 2023).

O termo jogos AAA é uma classificação usada na indústria de jogos para definir jogos com alto orçamento, com alto nível, e que normalmente são produzidos e distribuídos por grandes e conhecidas distribuidoras (*publishers*). Esses jogos costumam ser classificados como *blockbusters* devido à sua extrema popularidade. Muitos fazem parte de franquias de sucesso, com novos lançamentos sendo construídos no sucesso dos jogos anteriores (Arm, 2023).

2.1 Pontuação de Jogos

De acordo com a documentação da Steamworks (2023b), usuários com tempo de jogo em algum produto na Steam podem escrever uma análise (*review*), indicando se o recomendam ou não. Estas análises são uma forma fácil dos usuários compartilharem a experiência com o produto e descrever como o jogo ou software atendeu às expectativas. O valor agregado das análises positivas e negativas é usado para calcular uma pontuação exibida na página da loja, dando um indicativo de como os usuários analisaram o produto nos últimos 30 dias e desde o lançamento. Apenas análises de compras diretas pela Steam são levadas em conta para a pontuação agregada.

A Metascore é uma pontuação profissional criada pela Metacritic para filmes, jogos, programas de televisão e álbuns de música, que possuem pelo menos quatro análises publicadas. Esta métrica é calculada por meio de uma média ponderada de todas as críticas e avaliações das mídias especializadas, na qual é atribuída maior importância ou peso a alguns críticos e publicações do que a outros, baseado na sua qualidade e relevância. Além disso, a Metacritic também possui uma pontuação feita pelos usuários, uma vez que essa pontuação não é computada para o Metascore (Metacritic, 2023).

Segundo Park e Byun (2016), ao comprar um jogo, jogadores investigam diversos aspectos como trailer do jogo, capturas de tela, desenvolvedor, distribuidor, análises e pontuação dos jogos. As autoras estudaram a relação entre a pontuação profissional e a pontuação dos usuários da Metacritic em jogos da Steam, concluindo que não há correlação nas duas pontuações na maioria dos jogos. Adicionalmente, apenas jogos casuais e jogos independentes não tiverem diferença entre as pontuações. A pontuação desses jogos também é conhecida como *score review* ou *user score*.

2.1.1 Estimativa de Vendas

Em seu artigo, VG Insights (2021) estima quantas unidades de qualquer jogo da Steam foram vendidas utilizando a quantidade de *reviews* recebida, por meio de dados vazados de 11.445 jogos da Steam em 2018. Há mais de 90% de correlação entre as unidades vendidas e o número de *reviews*, e, além disso, a regressão simples múltipla utilizando o número de *reviews*, ano de lançamento e se é um jogo gratuito (*free to play*) como variáveis, obteve um R^2 ajustado de 78%.

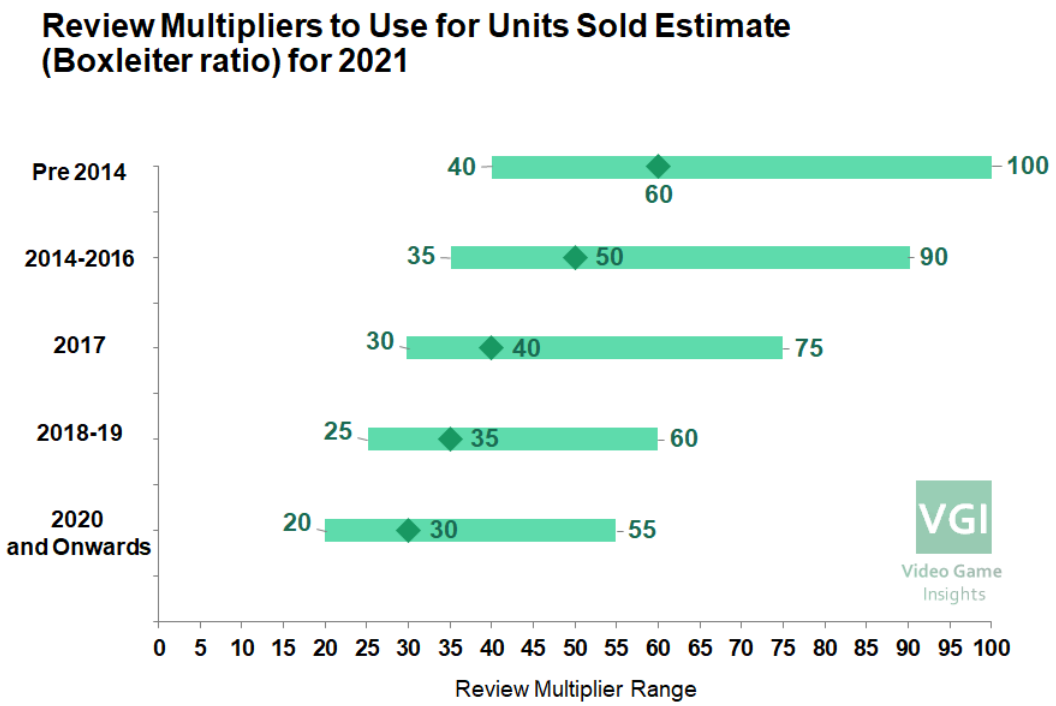


Figura 2.1: Multiplicadores de *reviews* a serem usados para estimativa de unidades vendidas para jogos da Steam por ano de lançamento (VG Insights, 2021).

Os multiplicadores de *reviews* para estimar as unidades vendidas caem consistentemente ao longo dos anos. Posto isso, os jogos lançados em 2020 têm cerca de 30 vezes mais avaliações e os jogos lançados até 2014 têm um multiplicador maior que 70 vezes. Dessa forma, é muito provável que os jogos lançados a partir de 2020 estejam na faixa de 20 a 50 vezes, enquanto que os jogos lançados antes de 2014 têm mais probabilidade de estarem na faixa de 40 a 100 vezes.

Para o conjunto de dados de jogos utilizados, esses intervalos acabaram sendo precisos para mais de 80% dos jogos, e o total estimado de unidades vendidas para os mais de 10.000 jogos analisados foi 7% maior do que as unidades reais vendidas. Ainda, jogos com preços mais altos tendem a ter múltiplos menores, MMOs têm múltiplos maiores, enquanto que esportes e corridas têm múltiplos menores do que a média.

2.2 Gêneros de Jogos

Ainda conforme a documentação da Steamworks (2023a), marcadores (*tags*) podem ser aplicados a um jogo pelo desenvolvedor, por jogadores e por moderadores da Steam. Isso permite que a comunidade ajude a rotular jogos com termos, temas e gêneros que auxiliam a descrever um jogo a outros jogadores. É exigido que jogos tenham cinco ou mais marcadores aplicados antes de serem lançados na plataforma, mas é recomendado que tenham até vinte marcadores. Os marcadores são divididos nas seguintes categorias:

- Principais Gêneros (*Top-Level Genres*);
- Gêneros (*Genres*);
- Subgêneros (*Sub-Genres*);
- Estilo Visual e Ponto de Vista (*Visuals and Viewpoint*);
- Temas e Atmosfera (*Themes and Moods*);
- Recursos (*Features*);
- Jogadores (*Players*);
- Outros Marcadores (*Other Tags*);
- Software (*Software*);
- Avaliações (*Assessments*);
- Avaliações e Outros (*Ratings etc*);
- Hardware/Entrada (*Hardware/Input*);
- Financiamento e Outros (*Funding etc.*).

Por exemplo, para a categoria de Principais Gêneros (*Top-Level Genres*) temos os seguintes marcadores: Ação (*Action*), Aventura (*Adventure*), Casual (*Casual*), Experimental (*Experimental*), Quebra-Cabeça (*Puzzle*), Corrida (*Racing*), RPG (*RPG*), Simulação (*Simulation*), Esportes (*Sports*), Estratégia (*Strategy*) e Jogos de Mesa (*Tabletop*).

Segundo Wulf et al. (2021), a variável gênero visa a identificar e a comparar diferentes jogos, principalmente em termos de diferenças de jogabilidade, ou seja, em termos de regras e de possibilidades de interação com um jogo. De acordo com Henry (2011), o gênero é destinado a transmitir informações sobre o jogo com base em suas características. Além disso, para o autor, os gêneros formam um sistema de categorização fundamental, permitindo que desenvolvedores e consumidores saibam o que esperar de um determinado jogo antes de desenvolvê-lo, comprá-lo ou jogá-lo.

2.3 Jogos Indie

Garda e Grabarczyk (2016) afirmam que, apesar da etimologia, o termo *indie* não é apenas uma abreviação do termo independente, mas também um rótulo para uma fase específica do fenômeno dos jogos independentes. Um jogo independente pode ser explicado como uma disjunção de três tipos de independência:

- Independência Financeira (relação desenvolvedor - investidor);
- Independência Criativa (relação desenvolvedor - público-alvo);
- Independência de Publicação (relação desenvolvedor - distribuidora).

Um jogo *indie* deve ser entendido como uma noção restrita que se refere apenas a um conjunto de jogos produzidos em um tempo e lugar específicos, ou seja, uma compreensão estreita e temporal dos jogos independentes. Além disso, esses jogos começaram a ser identificados por meio de algumas propriedades:

- Distribuição Digital;
- Natureza Experimental;
- Orçamento Pequeno e Preço Baixo;
- Estilo Retrô;
- Tamanho Pequeno;
- Time Pequeno;
- Mentalidade *Indie*;
- Cena *Indie*;
- *Middleware*.

No entanto, a Steam Web API permite a identificação de um jogo *indie* sem se restringir ao tipo de independência utilizada no seu desenvolvimento.

Ziyang (2021) definiu jogos *indie* (ou independentes) sob o domínio econômico, ou seja, considerou jogos *indie* aqueles jogos com um orçamento limitado em termos de processos de desenvolvimento, recursos humanos e publicidade. Ainda, segundo o autor, a maioria dos jogos *indie* possui um número de vendas relativamente baixo.

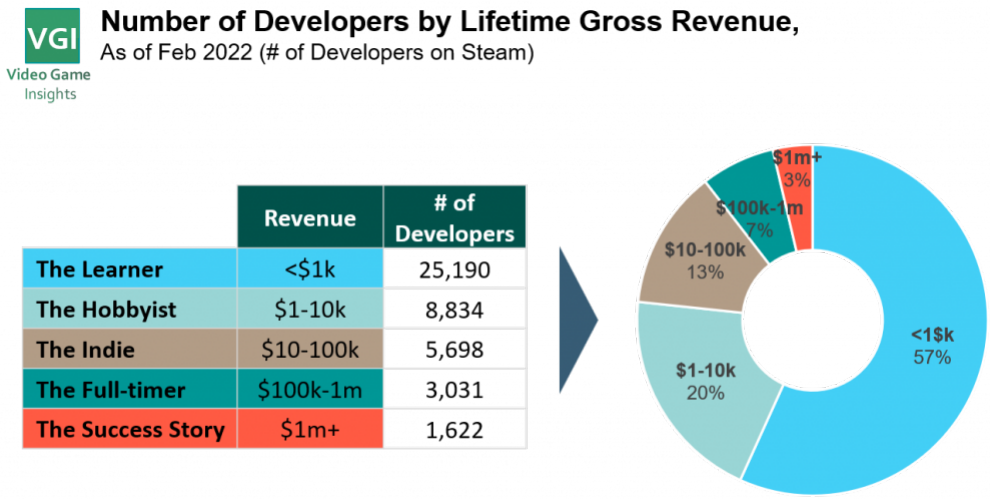
2.3.1 Sucesso no Desenvolvimento

Com o intuito de explorar o que desenvolvedores independentes que mais ganham na Steam estão fazendo, VG Insights (2022) analisou número de jogos desenvolvidos, autopublicação, especialidades, entre outros. Nota-se que apenas 10% dos desenvolvedores da Steam já ganharam mais de 100 mil dólares em receita bruta vitalícia.

Os desenvolvedores da Steam são classificados em 5 diferentes categorias com base em seus ganhos vitalícios na plataforma:

- $< \$$ 1k – *The Learner* – Desenvolvedores que normalmente lançam apenas 1 jogo e não é bom;
- \$ 1-10k – *The Hobbyist* – Desenvolvedores cujos jogos venderam algumas unidades, mas nunca se saíram muito bem;
- \$ 10-100k – *The Indie* – Este é o núcleo dos *indies*, eles ganharam algum dinheiro com seus jogos, mas normalmente não o suficiente para fazer disso uma carreira em tempo integral;

- $ 100k-1m – *The Full-timer* – Esses desenvolvedores ganharam o suficiente com seus jogos para que possam se tornar desenvolvedores *indie* em tempo integral;
- > $ 1m – *The Success History* – Esses desenvolvedores independentes conseguiram, ganharam mais de 1 milhão de dólares em receita bruta. Eles são tipicamente uma equipe, e não um desenvolvedor solo. Esse grupo de desenvolvedores se saiu bem e definitivamente está em minoria.



	Revenue	# of Developers
The Learner	<$1k	25,190
The Hobbyist	$1-10k	8,834
The Indie	$10-100k	5,698
The Full-timer	$100k-1m	3,031
The Success Story	$1m+	1,622

Figura 2.2: Número de desenvolvedores pela receita vitalícia (VG Insights, 2022).

Os desenvolvedores *The Success Story* possuem uma receita média por jogo muito maior do que os desenvolvedores menores ou menos sucedidos, fazendo em média mais de 3 milhões de dólares por jogo. Além disso, desenvolvedores bem-sucedidos também fazem mais jogos.

The Learners e *The Hobbyists*, conforme o artigo, normalmente fazem 1 a 2 jogos na Steam, e isso não necessariamente resulta em sucesso instantâneo, o que acaba em muitos novos desenvolvedores desistindo. *The Success Story* lançam em média de 4 a 5 jogos na Steam, e apesar de nem todos os jogos terem sucesso, eles continuam tentando.

	Average Revenue per game	Average # of games released
The Learner	$294	1.1
The Hobbyist	$2,275	1.6
The Indie	$14,999	2.4
The Full-timer	$111,172	3.0
The Success Story	$3,655,808	4.5

Figura 2.3: Número de jogos lançados e receita por jogo (VG Insights, 2022).

Cerca de 70 a 75% dos pequenos desenvolvedores *indie* publicam seus próprios jogos, enquanto que os mais sucedidos dividem igualmente entre publicar seu próprio jogo e ir com uma distribuidora (*publisher*). Bons desenvolvedores independentes são mais propensos a se concentrar no desenvolvimento e entrar em contato com uma distribuidora e bons jogos têm mais chances de chamar a atenção das distribuidoras, isso se deve ao fato do desenvolvedor já ter um histórico na indústria, tendo outros jogos publicados e ter trabalhado em grandes empresas de jogos.

Desenvolvedores de sucesso fazem muitos jogos, encontram distribuidoras para que possam se concentrar naquilo em que são bons, se concentram em gêneros que se saem bem, se adaptam às mudanças nas preferências dos jogadores, e também, se especializam em:

- Criação de recursos – Os desenvolvedores podem usar elementos de jogos anteriores, seja código ou base da arte, para simplificar significativamente o processo de desenvolvimento;
- Acúmulo de conhecimento – Os estúdios que se especializam aprendem mais sobre os detalhes do gênero;
- Construção de conexões – Concentrar-se em um tipo específico de jogo permite a construção de uma rede de suporte nesse espaço;
- Base de fãs – Os jogadores sabem o que recebem e isso é um bônus enorme.

3 Referencial Teórico

3.1 Machine Learning

Machine Learning (Aprendizagem de Máquina) envolve desenvolver programas que ajustam automaticamente seu desempenho de acordo com sua exposição às informações nos dados. Esse aprendizado é obtido por meio de um modelo parametrizado com parâmetros ajustáveis que são corrigidos automaticamente de acordo com diferentes critérios de desempenho (Igual e Seguí, 2017). Além disso, *Machine Learning* é considerada uma área da Inteligência Artificial (IA) e que pode ser dividida nas seguintes classes principais:

- Aprendizado supervisionado: Algoritmos que aprendem a partir de um conjunto de treino rotulado para generalizar para um conjunto com todas as entradas possíveis.
- Aprendizado não supervisionado: Algoritmos que aprendem a partir de um conjunto de treino não rotulado. Usado para explorar dados de acordo com algum critério estatístico, geométrico ou de similaridade.
- Aprendizagem por reforço: Algoritmos que aprendem por reforço a partir de recompensas e penalidades, fornecendo informações sobre a qualidade de uma solução, mas não sobre como melhorá-la. Melhores soluções são alcançadas explorando iterativamente o espaço de solução.
- Aprendizado semi-supervisionado: Algoritmos que podem melhorar automaticamente seu desempenho aprendendo a partir de um conjunto de treino não rotulado e sem interações externas (Igual e Seguí, 2017; Zhou, 2021).

Em outras palavras, a área de *Machine Learning* se preocupa com o desenvolvimento da capacidade de aprendizagem de máquina, e portanto, visa a criar teorias e procedimentos – algoritmos de aprendizagem – que permitam que as máquinas aprendam. Assim, *Machine Learning* é uma área interdisciplinar que combina resultados de estatística, lógica, robótica, ciência da computação, inteligência computacional, reconhecimento de padrões, mineração de dados, ciência cognitiva, entre outros (Chowdhary, 2020; Wojtusiak, 2012).

3.1.1 Modelos Lineares Generalizados

Modelos Lineares Generalizados (*Generalized Linear Models – GLM*) fornecem uma classe com aplicabilidade relativamente ampla e propriedades estatísticas desejáveis, estendendo características de um modelo linear para situações em que a variável resposta possui alguma distribuição da família exponencial.

Em um Modelo Linear Generalizado, Y tem distribuição na forma:

$$p(y_i \mid \theta_i, \alpha) = \exp\left(\frac{y_i\theta_i - b(\theta_i)}{\alpha} + c(y_i, \alpha)\right),$$

em que θ depende do preditor linear e o parâmetro de dispersão α é frequentemente conhecido. O modelo também possui uma função de ligação (*link function*) $g(\cdot)$, que fornece uma relação entre a média de y e o preditor linear.

Tabela 3.1: Tabela das distribuições GLM (Wakefield, 2013).

Distribuição	$\mathrm{N}(\mu, \sigma^2)$	$\mathrm{Poisson}(\mu)$	$\mathrm{Bernoulli}(\mu)$	$\mathrm{Gamma}\left(\frac{1}{\alpha}, \frac{1}{[\mu\alpha]}\right)$
Média $\mathrm{E}[Y \mid \theta]$	θ	$\exp(\theta)$	$\frac{\exp(\theta)}{1+\exp(\theta)}$	$-\frac{1}{\theta}$
Variância $V(\mu)$	1	μ	$\mu(1-\mu)$	μ^2
$b(\theta)$	$\frac{\theta^2}{2}$	$\exp(\theta)$	$\log(1+e^\theta)$	$-\log(-\theta)$
$c(y, \alpha)$	$-\frac{1}{2}\left[\frac{y^2}{2} + \log(2\pi\alpha)\right]$	$-\log y!$	1	$\frac{\log(y/\alpha)}{\alpha} - \log y + \log\Gamma(\alpha)$

A Tabela 3.1 mostra algumas características das principais distribuições dos modelos lineares generalizados, como informações sobre a média, a variância, as funções $b(\theta)$ e $c(y, \alpha)$. Além disso, a média é igual $E[Y] = \mu$ e a variância é igual a $var(Y) = \alpha V(\mu)$.

Para resumir, um Modelo Linear Generalizado assume uma relação linear em uma escala média transformada e uma forma da família exponencial para a distribuição da variável resposta. Assim, os modelos utilizam distribuições bem conhecidas, como as densidades Normal, Gama, Binomial e Poisson, mas inclui também outras distribuições mais atípicas, como a Normal Inversa e a Binomial Negativa. Isso amplia a aplicabilidade de ideias de modelos lineares a dados nas quais as variáveis respostas são positivas, contagens ou proporções, sem a necessidade de transformações (Wakefield, 2013; Davison, 2003).

3.1.2 Regressão Linear e Logística

Regressão Linear é um modelo para predizer uma variável resposta quantitativa com base em variáveis preditoras, em que se assume que existe aproximadamente uma relação linear entre as covariáveis e a variável resposta. Os coeficientes da regressão são desconhecidos e comumente são estimados utilizando a abordagem de mínimos quadrados. Para verificar se existe essa relação linear é utilizado um teste de hipóteses, em que a hipótese nula consiste em todos os coeficientes serem iguais a zero e a hipótese alternativa consiste em pelo menos um dos coeficientes serem diferentes de zero. O teste de hipóteses é realizado por meio da estatística F.

O modelo de Regressão Linear é dado por

$$\boldsymbol{Y} = \beta_0 + \beta_1 \boldsymbol{X_1} + \beta_2 \boldsymbol{X_2} + \cdots + \beta_p \boldsymbol{X_p} + \boldsymbol{\epsilon},$$

em que $\boldsymbol{X_j}$ representa o j-ésimo preditor, β_j é o coeficiente que quantifica a associação entre essa variável e a variável resposta, β_0 é o intercepto e $\boldsymbol{\epsilon}$ é um termo de erro aleatório com média zero. O β_j é interpretado como um efeito médio em $\boldsymbol{Y}$ de um aumento de uma unidade em $\boldsymbol{X_j}$, mantendo todos os outros preditores fixos.

Na Regressão Logística, em vez de modelar a resposta diretamente, modela-se a probabilidade da variável resposta pertencer a uma particular categoria utilizando a função logística, que garante saídas entre 0 e 1 para todos os valores das covariáveis. Os coeficientes da regressão são desconhecidos e são estimados utilizando o método da máxima verossimilhança.

O modelo utilizando a função logística é dado por

$$p(\boldsymbol{X}) = \frac{e^{\beta_0 + \beta_1 \boldsymbol{X_1} + \cdots + \beta_p \boldsymbol{X_p}}}{1 + e^{\beta_0 + \beta_1 \boldsymbol{X_1} + \cdots + \beta_p \boldsymbol{X_p}}}.$$

E, manipulando o modelo da função logística, obtemos

$$\frac{p(\boldsymbol{X})}{1 - p(\boldsymbol{X})} = e^{\beta_0 + \beta_1 \boldsymbol{X_1} + \cdots + \beta_p \boldsymbol{X_p}},$$

tal que $\frac{p(\boldsymbol{X})}{1-p(\boldsymbol{X})}$ é chamada de razão de chances e pode ter qualquer valor entre 0 e ∞. Valores de razão de chances perto de 0 e ∞ indicam muito baixas e muito altas probabilidades, respectivamente.

Além disso, após aplicarmos o logaritmo na razão de chances, obtemos o logito, que é linear nas covariáveis do modelo

$$\log\left(\frac{p(\boldsymbol{X})}{1 - p(\boldsymbol{X})}\right) = \beta_0 + \beta_1 \boldsymbol{X_1} + \cdots + \beta_p \boldsymbol{X_p}.$$

Em um modelo de Regressão Logística, como a relação de $p(\boldsymbol{X})$ e $\boldsymbol{X}$ não é linear, a quantidade que $p(\boldsymbol{X})$ muda devido a uma mudança de uma unidade em $\boldsymbol{X}$ dependerá do valor atual de $\boldsymbol{X}$. Mas, independentemente do valor de $\boldsymbol{X}$, se β for positivo, o aumento de $\boldsymbol{X}$ será associado ao aumento de $p(\boldsymbol{X})$, e se β for negativo, então o aumento de $\boldsymbol{X}$ estará associado à diminuição de $p(\boldsymbol{X})$ (James et al., 2013).

3.1.3 Regressão Ridge e Lasso

Regressão Ridge (*Ridge Regression*) é um método de encolhimento que busca reduzir a quantidade de variáveis independentes utilizando um estimador que diminui as estimativas dos coeficientes para zero. Essa abordagem é muito similar com o método de mínimos quadrados.

Em particular, as estimativas $\hat{\beta}^R$ dos coeficientes da Regressão Ridge são os valores que minimizam

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = SQRes + \lambda\sum_{j=1}^{p}\beta_j^2,$$

em que $\lambda \geq 0$ é um parâmetro de ajuste (*tuning parameter*), que é determinado separadamente, n é o número de observações e p é a quantidade de preditores.

Assim como nos mínimos quadrados, o parâmetro de Regressão Ridge busca estimativas de coeficientes que se ajustem bem aos dados, tornando a Soma dos Quadrados dos Resíduos (SQRes) pequena. No entanto, o termo $\lambda \sum_{j=1}^{p} \beta_j^2$, chamado de penalidade de encolhimento (*shrinkage penalty*), é pequeno quando $\beta_1, \cdots, \beta_p$ são próximos de zero e, portanto, tem o efeito de encolher (*shrinking*) as estimativas de β_j em direção a zero.

O parâmetro de ajuste λ serve para controlar o impacto relativo desses dois termos nas estimativas dos coeficientes de regressão. Quando $\lambda = 0$, o termo de penalidade não tem efeito e a Regressão Ridge produzirá as estimativas de mínimos quadrados. No entanto, quando λ tende ao infinito, o impacto da penalidade de encolhimento cresce e as estimativas dos coeficientes de Regressão Ridge se aproximarão de zero.

A Regressão Ridge incluirá todos os p preditores no modelo final, a penalidade irá encolher todos os coeficientes em direção a zero, porém não irá definir nenhum deles exatamente igual a zero, ou seja, não resultará em exclusão de nenhuma das variáveis. O Lasso é um método alternativo ao da Regressão Ridge, de forma que os coeficientes, $\hat{\beta}_\lambda^L$, minimizam a quantidade

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = SQRes + \lambda\sum_{j=1}^{p}|\beta_j|.$$

Assim como na Regressão Ridge, o Lasso encolhe as estimativas dos coeficientes em direção a zero. Porém, no caso do Lasso, a penalização tem o efeito de forçar algumas estimativas dos coeficientes para serem exatamente iguais a zero quando o parâmetro de ajuste λ é suficientemente grande. Portanto, o Lasso executa a seleção de variáveis e, como resultado, os modelos gerados a partir do Lasso são geralmente mais fáceis de interpretar do que os produzidos pela Regressão Ridge (James et al., 2013).

3.1.4 Árvores de Decisão

Árvores de Decisão (*Decision Trees*) podem ser aplicadas em ambos problemas de regressão e de classificação. Em regressão, para a criação das árvores, o espaço para os possíveis valores das covariáveis $X_1, \cdots, X_p$ é dividido em J regiões distintas e não sobrepostas, tal que este espaço forme retângulos de alta dimensão $R_1, R_2, \cdots, R_J$. Essas regiões são conhecidas como nós terminais ou folhas da árvore. O objetivo é encontrar regiões que minimizem a soma dos quadrados dos resíduos, dado por

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2,$$

em que $\hat{y}_{R_j}$ é a resposta média para as observações de treino dentro da j-ésima região.

Uma das estratégias adotadas é utilizar divisão binária recursiva para crescer uma árvore grande nos dados de treino, parando apenas quando cada nó terminal tiver menos do que um número mínimo de observações. Após isso, podá-la de volta para obter uma sequência das melhores subárvores, ou seja, árvores menores com menos divisões, mas que possam levar a uma maior redução na Soma dos Quadrados dos Resíduos (SQRes) posteriormente.

Para realizar a divisão binária recursiva é selecionado o preditor X_j e o ponto de corte s, de modo que dividir o espaço das covariáveis nas regiões $\{X|X_j < s\}$ e $\{X|X_j \geq s\}$ leva à maior redução possível na SQRes.

Mais detalhadamente, para quaisquer j e s, é definido o par de semiplanos

$$R_1(j,s) = \{X|X_j < s\} \quad \text{e} \quad R_2(j,s) = \{X|X_j \geq s\},$$

e são procurados os valores de j e s que minimizam a equação

$$\sum_{i:\, x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:\, x_i \in R_1(j,s)} (y_i - \hat{y}_{R_2})^2,$$

em que $\hat{y}_{R_1}$ é a resposta média para as observações de treino em $R_1(j,s)$ e $\hat{y}_{R_2}$ é a resposta média para as observações de treino em $R_2(j,s)$. Encontrar os valores de j e s que minimizam a equação pode ser feito rapidamente, principalmente quando o número de preditores p não é muito grande.

O processo é repetido buscando o melhor preditor e o melhor ponto de corte que dividam ainda mais os dados, de forma a minimizar a SQRes dentro de cada uma das regiões resultantes. O processo continua até que um critério de parada seja alcançado. Uma vez criadas as regiões $R_1, \cdots, R_J$, predizemos a resposta para uma determinada observação de teste usando a média das observações de treino na região à qual essa observação de teste pertence (James et al., 2013).

Em classificação, utilizamos as observações de treino para mapear a região em que cada observação de teste pertence e, em seguida, atribuímos à cada observação a classe mais comum nessa região. Ao interpretar seus resultados, estamos interessados tanto na predição da classe correspondente a uma determinada região do nó terminal, como também nas proporções de classe entre as observações de treino que caem nessa região.

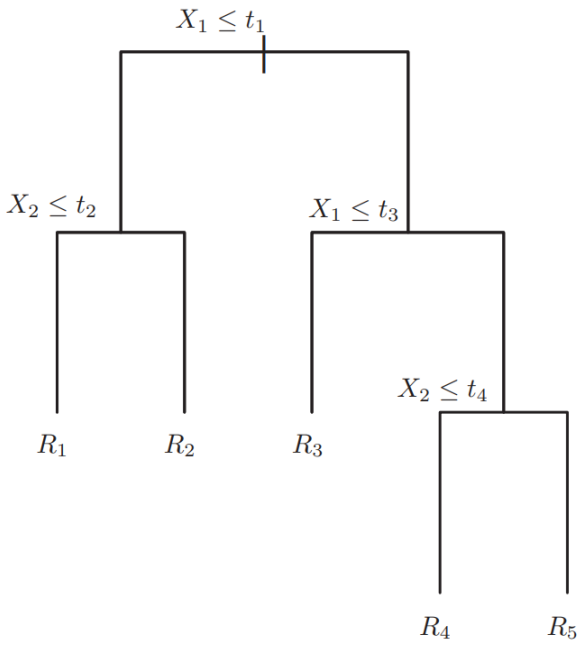


Figura 3.1: Representação de uma árvore particionada em 5 diferentes regiões utilizando o espaço de preditores X_1 e X_2 (James et al., 2013).

Assim como na regressão, é utilizada a divisão binária recursiva para crescer uma árvore de classificação. No entanto, como a Soma dos Quadrados dos Resíduos não pode ser utilizada como um critério para fazer as divisões binárias, uma alternativa é utilizar a taxa de erro de classificação. Como na classificação é atribuída uma observação em uma determinada região à classe de taxa de erro mais comum das observações de treino nessa região, a taxa de erro de classificação é a fração das observações de treino nessa região que não pertencem à classe mais comum. A taxa de erro de classificação é dada por

$$E = 1 - \max_k(\hat{p}_{mk}),$$

em que aqui $\hat{p}_{mk}$ representa a proporção das observações de treino na m-ésima região que são da k-ésima classe. No entanto, a taxa de erro de classificação não é suficientemente sensível para o crescimento de árvores e outras medidas são preferíveis, sendo uma delas o Índice Gini, definido por:

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}),$$

em que $\hat{p}_{mk}$ também representa a proporção das observações de treino na m-ésima região que são da k-ésima classe.

O Índice Gini é uma medida da variância total entre as K classes. Ele assume um valor pequeno se todos os $\hat{p}_{mk}$'s estão próximos de zero ou um, e por essa razão, o índice Gini refere-se a uma medida de pureza do nó (*node purity*) – um valor pequeno indica que um nó contém predominantemente observações de uma única classe (James et al., 2013).

Há vários motivos pelos quais as árvores de decisão pequenas são preferidas, sendo uma delas a interpretabilidade. Um especialista humano pode achar simples analisar, explicar e talvez até corrigir uma árvore de decisão que consiste em não mais do que alguns testes. Quanto maior a árvore, mais difícil é. Outra vantagem das árvores de decisão pequenas é a tendência de descartar informações irrelevantes e redundantes.

Finalmente, árvores maiores são propensas a sobreajustar os exemplos de treino. Isso ocorre porque o método de dividir e conquistar continua dividindo o conjunto de treino em subconjuntos cada vez menores, sendo o número dessas divisões igual ao número de testes de atributo na árvore. Por último, os subconjuntos de treino resultantes podem se tornar tão pequenos que as classes podem ser separadas por um atributo que apenas por acaso (ou ruído) tem um valor diferente nos exemplos positivos e negativos restantes (Kubat, 2017).

3.1.5 Bagging

As árvores de decisão sofrem de alta variância, isso significa que, se dividirmos os dados de treino em duas partes aleatoriamente e ajustarmos uma árvore de decisão em ambas as metades, os resultados obtidos poderão ser bem diferentes. Uma alternativa é o *Bagging*, que utiliza *bootstrap* para tomar amostras repetidas do conjunto de dados de treino. Nesta abordagem, geramos B diferentes conjuntos de dados de treino *bootstrap*, treinamos nosso método no b-jésimo conjunto para conseguir $\hat{f}^{*b}(x)$ e, por fim, calculamos a média de todas as previsões para obter

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x).$$

Em regressão, B árvores são construídas utilizando B conjuntos de dados de treino *bootstrap* e é calculada a média das predições resultantes. Como essas árvores crescem profundamente e não são podadas, cada árvore possui alta variância e baixo viés, e, portanto, a variância é reduzida calculando a média dessas B árvores. *Bagging* tem demonstrado melhorias impressionantes na precisão, combinando centenas e milhares de árvores em um único procedimento.

Em classificação, para uma determinada observação de teste, podemos registrar a classe prevista por cada uma das B árvores e a previsão geral é a classe majoritária que ocorre mais comumente entre as B previsões. O número de árvores não é um parâmetro crítico com *Bagging*, usar um valor muito grande de B não levará a um sobreajuste. Na prática, usamos um valor de B suficientemente grande para que o erro se estabeleça.

Embora a coleção de árvores *Bagging* seja muito mais difícil de interpretar do que uma única árvore, pode-se obter um resumo geral da importância de cada preditor usando a Soma dos Quadrados dos Resíduos (regressão) ou o índice Gini (classificação). Em ambos os casos, a quantidade total da Soma dos Quadrados dos Resíduos ou o valor total do índice Gini é diminuído devido a divisões em um determinado preditor, com a média de todas as B árvores, tal que um valor grande indica um preditor importante (James et al., 2013).

3.1.6 Random Forest

Random Forests (Florestas Aleatórias) fornecem uma melhoria em relação às árvores *Bagging* por meio de um pequeno ajuste aleatório que descorrelaciona as árvores. Assim como no *Bagging*, construímos uma floresta de árvores de decisão em amostras de treino *bootstrap*, mas ao construir essas árvores, cada vez que uma divisão em uma árvore é considerada, uma amostra aleatória de m preditores é escolhida como candidatos a divisão do conjunto completo de p preditores. A divisão permite usar apenas um desses m preditores e uma nova amostra de m preditores é retirada em cada divisão, tal que o número de preditores considerados em cada divisão é aproximadamente igual à raiz quadrada do número total de preditores, ou seja, $m \approx \sqrt{p}$.

A principal diferença entre o *Bagging* e as *Random Forests* é a escolha do subconjunto preditor de tamanho m. Por exemplo, se uma floresta aleatória é construída usando $m = p$, isso equivale simplesmente ao *Bagging*. Usar um valor pequeno de m na construção de uma floresta aleatória normalmente será útil quando tivermos um grande número de preditores correlacionados (James et al., 2013).

3.2 Medidas de Desempenho

Para avaliar a capacidade de generalização dos modelos, precisamos não apenas de métodos de estimação práticos e eficazes, mas também de algumas medidas de desempenho que possam quantificar a capacidade e a qualidade de um modelo, que depende do algoritmo e dos dados. Ou seja, precisamos quantificar até que ponto o valor de resposta previsto para uma determinada observação está próximo do valor de resposta real para essa observação. Fundamentalmente, o processo de avaliação tenta verificar qual modelo produz as previsões mais precisas e úteis (Zhou, 2021; James et al., 2013; Bruce et al., 2020).

3.2.1 Medidas de Desempenho em Regressão

As principais medidas de desempenho utilizadas em regressão:

- Erro Quadrático Médio (*Mean Squared Error* - MSE):
  O Erro Quadrático Médio de um modelo em relação a um conjunto de teste é a média dos erros quadrados de previsão em todas as observações no conjunto de teste. O erro de previsão é a diferença entre o valor verdadeiro e o valor previsto para uma observação. O MSE também pode ser utilizado se houver *outliers* que precisam ser detectados, atribuindo pesos maiores a esses pontos. Se o modelo eventualmente produzir uma única previsão muito ruim, a parte quadrada da função aumenta o erro.

  $$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2,$$

  em que y_i é o verdadeiro valor para a observação de teste x_i, $\hat{f}(x_i)$ é o valor da predição para a observação de teste x_i e n é o número de observações.

- Raiz do Erro Quadrático Médio (*Root Mean Squared Error* - RMSE):
  A Raiz do Erro Quadrático Médio mede o erro de previsão médio feito pelo modelo ao prever o resultado de uma observação. Ou seja, a diferença média entre os valores de resultados conhecidos observados e os valores previstos pelo modelo. Quanto menor o RMSE, melhor o modelo.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2}.$$

- Erro Médio Absoluto (*Mean Absolute Error* - MAE):
  O Erro Absoluto Médio de um modelo em relação a um conjunto de teste é a média dos valores absolutos dos erros de previsão individuais em todas as observações do conjunto de teste. Cada erro de previsão é a diferença entre o valor verdadeiro e o valor previsto para a observação. O MAE, que corresponde à diferença média absoluta entre os resultados observados e previstos, é menos sensível a *outliers*. Assim como no MSE e no RMSE, quanto menor o valor do MAE, melhor o modelo.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{f}(x_i)|.$$

Mais informações sobre as métricas podem ser encontradas em Sammut e Webb (2017), Kassambara (2018), Chicco et al. (2021) e James et al. (2013).

3.2.2 Medidas de Desempenho em Classificação

As principais medidas de desempenho utilizadas em classificação:

- Acurácia (*Accuracy*):
  A Acurácia é a proporção de observações classificadas corretamente, e que é diretamente relacionada com a taxa de erro de classificação,

$$Acc = 1 - E.$$

- *Kappa* (κ):
  O *Kappa* mede a possibilidade de uma previsão correta gerada apenas pelo acaso. A estatística pode ter valores entre -1 e 1, quando $\kappa = 1$ temos uma concordância perfeita entre a predição observada e a predição esperada, enquanto que quando $\kappa = 0$ não temos uma concordância entre o observado e as classes preditas.

$$\kappa = \frac{P(o) - P(e)}{1 - P(e)},$$

  em que $P(o)$ e $P(e)$ denotam a probabilidade de concordância observada e esperada entre o que foi classificado e os valores verdadeiros.

- Matriz de Confusão (*Confusion Matrix*):

  A Matriz de Confusão resume o desempenho de classificação em relação aos dados de teste. É uma matriz bidimensional, indexada em uma dimensão pela verdadeira classe de uma variável e na outra pela classe que o classificador atribui.

  Um caso particular da Matriz de Confusão é com uma variável de duas classes, uma designada como classe positiva e a outra como classe negativa. Nesse contexto, as quatro células da matriz são designadas como Verdadeiros Positivos (TP), Falsos Positivos (FP), Verdadeiros Negativos (TN) e Falsos Negativos (FN) (Ting, 2017).

		Assigned class	
		Positive	**Negative**
Actual class	Positive	True positive (TP)	False negative (FN)
	Negative	False positive (FP)	True negative (TN)

Figura 3.2: Representação de uma matriz de confusão (Ting, 2017).

- Sensibilidade (*Sensitivity*):
  A Sensibilidade, também conhecida como Taxa de Verdadeiros Positivos, mede a proporção das classes positivas que foram corretamente classificadas.

$$Sensitivity = \frac{TP}{TP + FN}.$$

- Especificidade (*Specificity*):
  A Especificidade, também conhecida como Taxa de Verdadeiros Negativos, mede a proporção das observações negativas que foram corretamente classificadas.

$$Specificity = \frac{TN}{TN + FP}.$$

Mais informações sobre as métricas podem ser encontradas em Kubat (2017), Sammut e Webb (2017), Dinov (2018), Kuhn et al. (2013) e Hossin e M.N (2015).

3.3 Web API

Uma API (*Application Programming Interface*) ou Interface de Programação de Aplicativo é um conjunto de funções de software pelas quais um aplicativo pode fazer requisições de um serviço de software de nível inferior, biblioteca ou sistema operacional. É uma maneira de um software pedir a outro software para fazer algo. Por exemplo, no caso de chamadas do sistema operacional, o aplicativo pode solicitar funções básicas, como acesso ao sistema de arquivos (Li e Jain, 2009).

As Web APIs, também chamadas de serviços RESTful, quando em conformidade com os princípios de arquitetura REST, são caracterizadas por sua relativa simplicidade e sua adequação natural para a Web. As Web APIs contam quase inteiramente com o uso de URIs para identificação e interação de recursos, e com HTTP para transmissão de mensagem (Maleshkova et al., 2010).

Um dos recursos de uma arquitetura RESTful é o uso de códigos de resposta HTTP. Se uma requisição é enviada a um servidor e se não houver nenhum problema, provavelmente será retornado um código de resposta HTTP de 200 ("OK"). Se algo der errado, o código de resposta estará no intervalo 3xx, 4xx ou 5xx: por exemplo, 500 ("Erro Interno do Servidor"). Um código de resposta de erro é um sinal para o cliente de que os metadados e o corpo da entidade não devem ser interpretados como uma resposta à requisição. Ou seja, não é o que o cliente pediu: é a tentativa do servidor de informar o cliente sobre um problema (Richardson e Ruby, 2008).

Conforme HTTP Statuses (2023), os códigos de resposta HTTP são divididos em 5 grupos:

- 1xx – Informativo (*Informational*)
- 2xx – Sucesso (*Success*)
- 3xx – Redirecionamento (*Redirection*)
- 4xx – Erro do Cliente (*Client Error*)
- 5xx – Erro do Servidor (*Server Error*)

E alguns dos códigos de resposta HTTP mais comuns são:

- 200 – OK – A requisição foi sucedida.
- 403 – Proibido (*Forbidden*) – O servidor entendeu a requisição, porém se recusou a autorizá-la.
- 429 – Muitas Requisições (*Too Many Requests*) – O usuário enviou muitas requisições em um determinado período de tempo (limitação de taxa).
- 503 – Serviço Indisponível (*Service Unavailable*) – O servidor está atualmente indisponível para lidar com a requisição devido a uma sobrecarga temporária ou manutenção programada, e provavelmente será liberado após algum atraso.

Por fim, segundo Richardson e Ruby (2008), um número considerável das requisições retorna estruturas de dados simples (números, matrizes, *hashes*, e assim por diante), serializadas como strings formatadas em JSON. Portanto, é possível realizar requisições utilizando o método HTTP GET e obter, com as Web APIs, estruturas de dados nesse formato.

4 Metodologia

4.1 Coleta dos Dados

Os dados foram coletados através da Steam Web API utilizando a linguagem de programação Python, como também a IDE (*Integrated Development Environment*) VS Code. Um *script* foi desenvolvido para a coleta dos dados. Cada observação da base de dados se refere a um aplicativo (*app*) da Steam, de forma que um aplicativo pode se referir a um jogo, demo, pacote de expansão, entre outros. Além disso, os dados foram obtidos utilizando os seguintes passos:

- Obtenção da lista de aplicativos da Steam;
- Obtenção de informações referentes a cada aplicativo;
- Obtenção das avaliações de usuários de cada aplicativo.

O resultado da coleta foi 159.260 arquivos *.json*, de forma que cada arquivo contém informações de cada aplicativo, identificados pelo seu respectivo *app_id*. Foi necessária cerca de uma semana de execução para a obtenção de todos os dados.

4.2 Processamento dos Dados

O processamento, a análise e a modelagem dos dados foram realizadas na IDE RStudio com a linguagem de programação R. No processamento foram iterados todos os arquivos *.json* para tabular a base de dados, foram também construídas novas variáveis a partir das informações disponíveis e, por fim, foi criado um *dataframe* com a seguinte estrutura:

- app_id: Número de identificação do aplicativo;
- name: Nome do aplicativo;
- type: Tipo de aplicativo – jogo, demo, dlc, etc;
- developers: Desenvolvedores;
- publishers: Distribuidoras;
- plataforms: Quantidade de plataformas em que está disponibilizado – 1 a 3;
- supported_languages: Quantidade de idiomas disponíveis;
- required_age: Idade mínima para acesso;
- is_free: Se o aplicativo é gratuito ou não;
- price_overview: Preço do aplicativo;
- categories: Categorias do aplicativo;
- singleplayer: Se possui opção com um único jogador;

- multiplayer: Se possui opção com múltiplos jogadores;
- is_online: Se possui opção de ser jogado online;
- controller_support: Se possui compatibilidade com o controle;
- trading_cards: Se possui cartas colecionáveis Steam;
- workshop: Se possui oficina Steam;
- genres: Gênero de jogos;
- indie: Se o aplicativo é *Indie*;
- early_access: Se o aplicativo foi lançado em acesso antecipado;
- dlc: Quantidade de DLCs (*Downloadable Content*);
- achievements: Quantidade de conquistas Steam;
- coming_soon: Se o aplicativo ainda não foi lançado;
- date: Data de lançamento;
- release_date: Data de lançamento formatada;
- period_time: Período pré-pandêmico, pandêmico e pós-pandêmico;
- adult_content: Se possui conteúdo exclusivamente adulto;
- metacritic: Pontuação na Metacritic;
- total_positive: Total de avaliações positivas dos usuários;
- total_reviews: Total de avaliações dos usuários;
- score_desc: Descrição das avaliações;
- user_score: Proporção de positivos do total de avaliações.

4.3 Segmentação e Análise dos Dados

Mesmo com o trabalho tendo como foco jogos *Indie* de 2018 a 2022, foram feitas algumas análises iniciais, assim como alguns filtros para termos a base de interesse. Em relação ao tipo de aplicativo, a Tabela 4.1 mostra que aproximadamente metade dos aplicativos da base de dados inicial é composta por jogos (53.49%) e um quarto por dlc (25.01%).

Tabela 4.1: Quantidade e Percentual – Tipo de Aplicativo.

Aplicativo	Quantidade	Percentual
game	85 195	53.49%
dlc	39 829	25.01%
NA	13 952	8.76%
demo	9 689	6.08%
music	4 846	3.04%
episode	2 651	1.66%
movie	1 734	1.09%
video	946	0.59%
advertising	216	0.14%
mod	97	0.06%
series	96	0.06%
hardware	9	0.01%
Total	159 260	100%

Em seguida, os seguintes filtros foram utilizados para segmentar a base, como também para fazer uma análise da descrição da pontuação desses jogos:

- type: tipo de aplicativo filtrado em jogos (*game*);
- indie: jogos filtrados para serem *indie*;
- release_date: data de lançamento do jogo filtrada entre 2018 e 2022;
- adult_content: não serão utilizados jogos com conteúdo exclusivamente adulto.

Jogos recebem uma descrição de avaliação positiva, negativa ou neutra quando possuem pelo menos 10 avaliações feitas por usuários. A Tabela 4.2 nos mostra que 20% dos jogos *indie* lançados entre 2018 e 2022 não possuem nenhuma avaliação, indicando que uma parte considerável dos jogos *indie* lançados na Steam são testes ou pequenos projetos, e que não possuem um comprometimento real de comercialização ou de um produto completo com uma alta qualidade.

Tabela 4.2: Quantidade e Percentual – Descrição das Avaliações desses Jogos.

Descrição Avaliações	Quantidade	Percentual
No user reviews	6 489	19.42%
Positive	4 642	13.90%
Very Positive	3 782	11.32%
Mixed	3 018	9.03%
1 user reviews	2 926	8.76%
Mostly Positive	2 481	7.43%
2 user reviews	2 239	6.70%
3 user reviews	1 640	4.91%
4 user reviews	1 364	4.08%
5 user reviews	1 131	3.39%
6 user reviews	901	2.70%
7 user reviews	741	2.22%
8 user reviews	655	1.96%
9 user reviews	513	1.54%
Mostly Negative	401	1.20%
Overwhelmingly Positive	382	1.14%
Negative	98	0.29%
Very Negative	2	0.01%
Overwhelmingly Negative	1	0%
NA	1	0%
Total	33 407	100%

Além disso, a distribuição da quantidade de avaliações feitas por usuários na Tabela 4.3 mostra que metade desses jogos tem até 6 avaliações, como também que 86.39% possuem até 100 avaliações. Logo, é possível perceber que poucos desses jogos conseguem ter um número expressivo de análises, e portanto, estaremos considerando apenas aqueles com pelo menos 5 avaliações recebidas, formando uma amosta com 18.748 observações.

Tabela 4.3: Descritivas – Avaliações na Steam – Jogos *Indie* de 2018 a 2022.

Reviews	n	Mínimo	1º Quartil	Mediana	Média	3º Quartil	Máximo	Desvio
Total	33 407	0	1	7	441	30	730 500	7 926
Máximo 1k	32 186	0	1	6	45	24	999	121
Máximo 100	28 860	0	1	4	13	15	100	19

Gêneros dos Jogos *Indie*

Para a base final com 18.748 observações foi realizada uma análise mais aprofundada dos gêneros desses jogos, pois um jogo pode ter diversos gêneros conjuntamente. Os principais objetivos dessa análise foram entender quais são os gêneros mais utilizados nos jogos *indie*, mostrar a associação desses grupos e, por fim, realizar um agrupamento para ser utilizado na modelagem.

Tabela 4.4: Quantidade e Percentual – Gêneros dos Jogos *Indie*.

Gêneros	Quantidade	Percentual
Adventure	8 670	21.52%
Casual	8 209	20.37%
Action	8 178	20.30%
Simulation	4 463	11.08%
Strategy	3 872	9.61%
RPG	3 792	9.41%
Sports	866	2.15%
No Genre	849	2.11%
Racing	719	1.78%
Massively Multiplayer	411	1.02%
Violent	87	0.22%
Gore	55	0.14%
Nudity	26	0.06%
Sexual Content	22	0.05%
Utilities	15	0.04%
Education	12	0.03%
Audio Production	10	0.02%
Animation & Modeling	8	0.02%
Design & Illustration	7	0.02%
Game Development	7	0.02%
Video Production	5	0.01%
Software Training	4	0.01%
Photo Editing	2	0%
Accounting	1	0%
Web Publishing	1	0%
Total	40 291	100%

Os gêneros de Aventura, Casual e Ação são os que possuem maior representação dentro dos jogos, de forma que cada um deles represente em torno de 20% do total. Além disso, pela Tabela 4.4, diversos gêneros são pouco utilizados, aparecendo em menos de 1% das vezes.

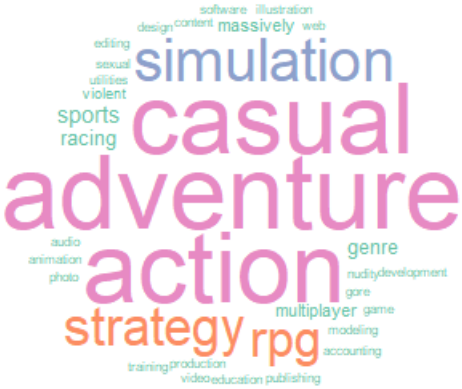


Figura 4.1: Visualização dos gêneros por uma nuvem de palavras.

Base de Treino e de Teste

A base de dados utilizada é composta por jogos *indie* lançados entre 2018 e 2022. A divisão para treino e teste foi realizada utilizando uma divisão temporal, ou seja, a base de treino foi composta por jogos lançados entre 2018 e 2021, enquanto que a base de teste foi composta por jogos lançados em 2022. Assim, temos uma divisão de treino com 14.663 observações (0.782) e de teste com 4.085 (0.218).

A divisão por intervalos de anos foi realizada uma vez que temos um efeito temporal na quantidade de avaliações. Ou seja, jogos que foram lançados a mais tempo possuem um período maior para receberem avaliações dos usuários. Além disso, existem certas limitações no banco de dados, até o momento da coleta os jogos não possuíam uma data de saída do acesso antecipado na Steam Web API. Portanto, uma seleção temporal nos permite uma melhor visualização da variável resposta ao longo do tempo.

5 Resultados

Gêneros dos Jogos *Indie*

A técnica de Regras de Associação foi utilizada para verificar a relação dos gêneros dos jogos *indie*, ou seja, para verificar o suporte (*support*), a confiança (*confidence*) e o *lift* para os conjuntos de gêneros.

Dado um item A, um item B e a regra $\{A\} \longrightarrow \{B\}$, temos que

- suporte: representa a probabilidade de A e B ocorreram simultaneamente em relação ao total;
- confiança: representa a probabilidade de B ocorrer dado que A ocorreu;
- *lift*: mede o quanto mais frequente torna-se B quando A ocorre.

Informações adicionais sobre regras de associação podem ser vistas em de Vasconcelos e de Carvalho (2018) e Gonçalves (2005).

A Tabela 5.1 mostra o resultado da aplicação das regras de associação, e podemos obter algumas conclusões:

- Aventura, Casual e Ação possuem probabilidade de ocorrer maior que 0.43.
- Ação e Aventura possuem probabilidade igual a 0.2253 de aparecerem conjuntamente.
  - Dado que o jogo possua o gênero de Ação, existe uma probabilidade de 0.5164 do jogo também ter Aventura.
  - A relação possui um *lift* maior que 1, indicando que um dos gêneros se torna mais frequente quando o outro ocorre.
- RPG e Ação possuem probabilidade igual a 0.0863 de aparecerem conjuntamente.
  - Dado que o jogo possua o gênero de RPG, existe uma probabilidade de 0.4267 do jogo também ter Ação.
  - A relação possui um *lift* menor que 1, indicando que um dos gêneros se torna menos frequente quando o outro ocorre.

Tabela 5.1: Regras de Associação para Gêneros – Top 25.

Regras	Suporte	Confiança	Lift	Quantidade
{} ⟶ {Adventure}	0.4624	0.4624	1.0000	8 670
{} ⟶ {Casual}	0.4379	0.4379	1.0000	8 209
{} ⟶ {Action}	0.4362	0.4362	1.0000	8 178
{} ⟶ {Simulation}	0.2381	0.2381	1.0000	4 463
{Action} ⟶ {Adventure}	0.2253	0.5164	1.1166	4 223
{Adventure} ⟶ {Action}	0.2253	0.4871	1.1166	4 223
{} ⟶ {Strategy}	0.2065	0.2065	1.0000	3 872
{} ⟶ {RPG}	0.2023	0.2023	1.0000	3 792
{Casual} ⟶ {Adventure}	0.1815	0.4145	0.8964	3 403
{Adventure} ⟶ {Casual}	0.1815	0.3925	0.8964	3 403
{Action} ⟶ {Casual}	0.1492	0.3421	0.7814	2 798
{Casual} ⟶ {Action}	0.1492	0.3408	0.7814	2 798
{RPG} ⟶ {Adventure}	0.1353	0.6690	1.4467	2 537
{Adventure} ⟶ {RPG}	0.1353	0.2926	1.4467	2 537
{Simulation} ⟶ {Casual}	0.1271	0.5339	1.2194	2 383
{Casual} ⟶ {Simulation}	0.1271	0.2903	1.2194	2 383
{Strategy} ⟶ {Casual}	0.1007	0.4873	1.1130	1 887
{Casual} ⟶ {Strategy}	0.1007	0.2299	1.1130	1 887
{Simulation} ⟶ {Adventure}	0.0928	0.3899	0.8431	1 740
{Adventure} ⟶ {Simulation}	0.0928	0.2007	0.8431	1 740
{RPG} ⟶ {Action}	0.0863	0.4267	0.9782	1 618
{Action} ⟶ {RPG}	0.0863	0.1978	0.9782	1 618
{Strategy} ⟶ {Simulation}	0.0857	0.4150	1.7434	1 607
{Simulation} ⟶ {Strategy}	0.0857	0.3601	1.7434	1 607
{Simulation} ⟶ {Action}	0.0798	0.3352	0.7684	1 496

Por fim, para os agrupamentos de gêneros foi utilizado Agrupamento Hierárquico (*Hierarchical Clustering*), uma técnica de aprendizado não-supervisionado. As variáveis numéricas *plataforms*, *supported_languages*, *required_age*, *price_overview*, *dlc* e *achievements* foram usadas e padronizadas para calcular a distância euclidiana dos gêneros, que foram agrupados em 5 grupos para o dendograma e para serem utilizados na base de modelagem.

Portanto, os 5 grandes grupos de gêneros ficaram como:

- Action & Adventure;
- Casual;
- RPG & Strategy;
- Simulation;
- Sports & Others.

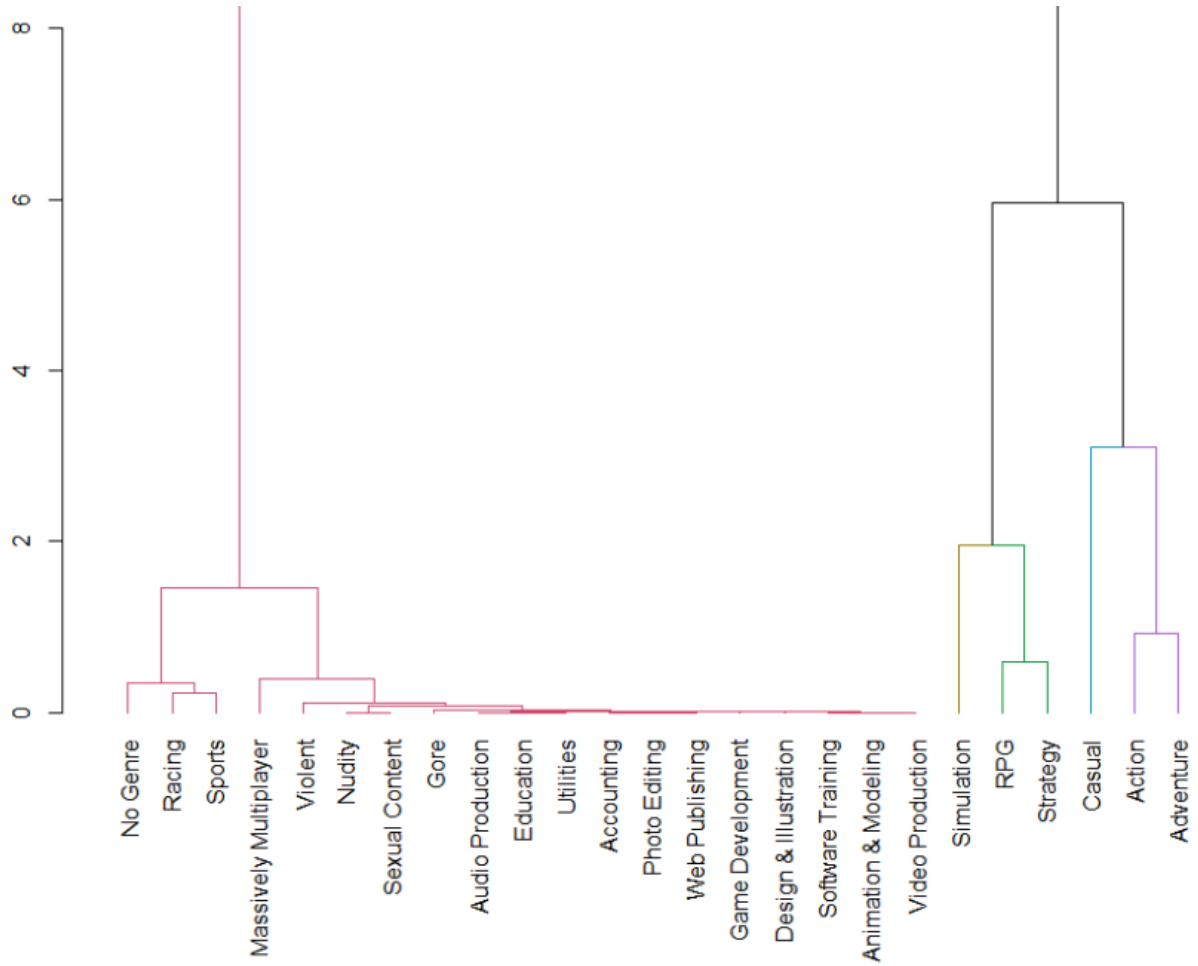


Figura 5.1: Dendograma com o agrupamento para 5 gêneros.

5.1 Steam Análise

A análise foi realizada para entender as distribuições das 3 variáveis respostas e suas relações com as covariáveis na base de treino. As variáveis respostas serão utilizadas como um indicativo da popularidade dos jogos. Portanto, teremos uma variável numérica de contagem, uma variável contínua entre 0 e 1 e uma variável categórica.

- total_reviews: Total de avaliações dos usuários;
- user_score: Proporção de positivos do total de avaliações;
- new_score: Descrição das avaliações.

Conforme visto anteriormente, a base de dados utilizada é composta por jogos *indie* lançados entre 2018 e 2022, de forma que a base de treino é composta por jogos lançados entre 2018 e 2021, enquanto que a base de teste é composta por jogos lançados em 2022.

Como já foi mencionado, jogos recebem uma descrição de avaliação positiva, negativa ou neutra quando possuem pelo menos 10 avaliações feitas por usuários. Ainda, um jogo é classificado com uma descrição positiva quando possui uma proporção de positivos de pelo menos 0.7.

Dado que estamos utilizando jogos *indie* com pelo menos 5 avaliações, a variável *new_score* foi refeita para que se considerasse o *user_score* dos jogos que não tinham a descrição da avaliação. A nova descrição possui dois níveis (positivos e não positivos), uma vez que temos poucos jogos com descrições negativas.

O agrupamento dos gêneros nos permite entender o comportamento dos jogos dentro de cada grupo. Portanto, além de fazermos uma análise da base geral, também faremos para cada grupo de gênero para compararmos os comportamentos.

Tabela 5.2: Quantidade e Percentual – New Score – Base de Treino.

New Score	Quantidade	Percentual
Positive	10 435	71.17%
Non-Positive	4 228	28.83%
Total	14 663	100%

Tabela 5.3: Quantidade e Percentual – New Score – Abertura por Gêneros.

Grupos de Gêneros	Quantidade	Positive	Non-Positive
Action & Adventure	9 948	70.76%	29.24%
Casual	6 458	73.20%	26.80%
RPG & Strategy	5 059	66.26%	33.74%
Simulation	3 435	60.12%	39.88%
Sports & Others	2 051	65.43%	34.57%

Pela Tabela 5.2, 71.17% dos jogos *indie* possuem classificação positiva. Porém, conforme a Tabela 5.3, conseguimos ver diferenças nessa proporção de positivos entre os diferentes gêneros. Jogos que possuem o gênero Casual são os que têm maior proporção de classificações positivas, conseguindo 73.20% com essa classificação. Por outro lado, jogos que possuem o gênero de Simulação são os que têm menor proporção, alcançando apenas 60.12% desses jogos.

Figura 5.2: Distribuição – New Score – Base de Treino.

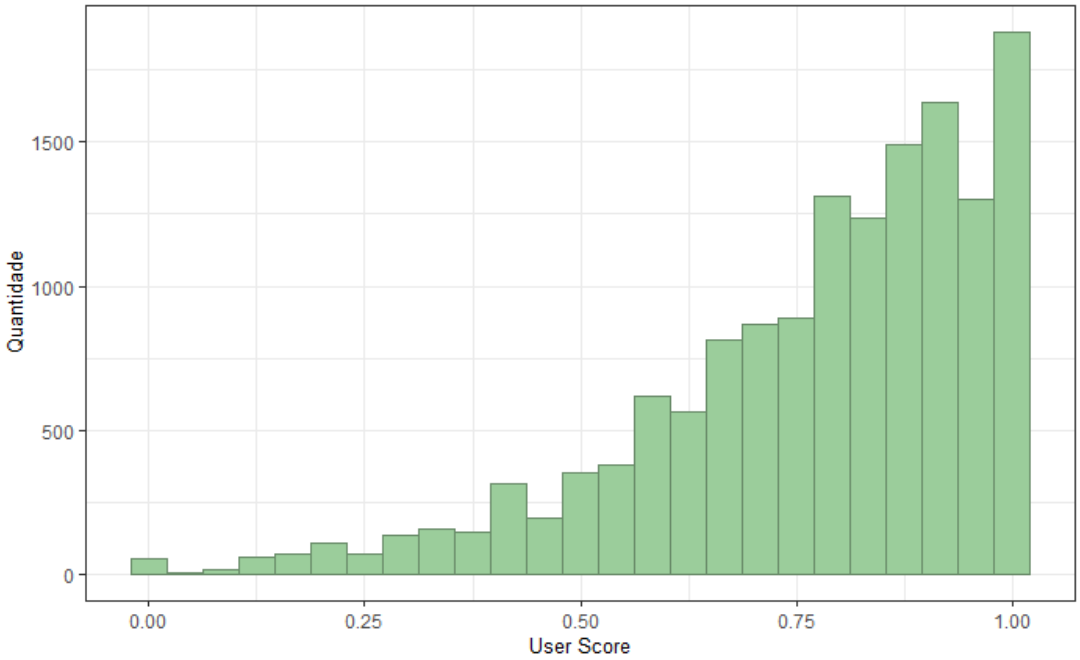


Figura 5.3: Distribuição – User Score – Base de Treino.

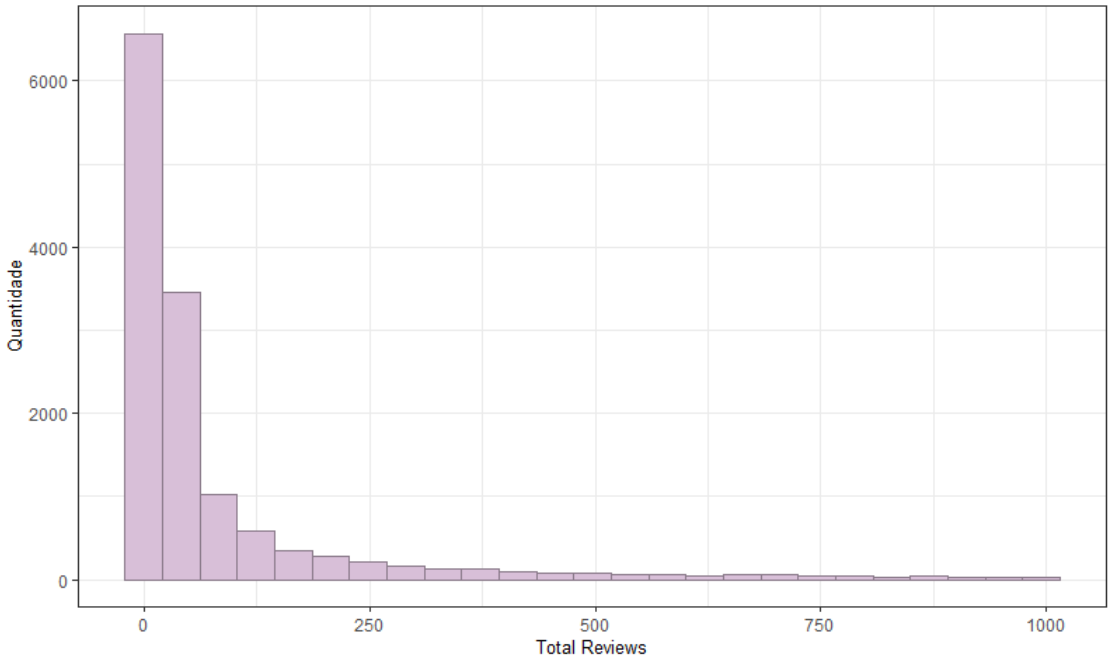


Figura 5.4: Distribuição – Total Reviews – Jogos com no máximo 1000 avaliações.

Na Figura 5.5, podemos observar o gráfico de correlação de Spearman para as variáveis numéricas da base de treino. Em relação às variáveis respostas, *user_score* teve maior correlação positiva com *achievements*; *total_reviews* teve maior correlação positiva com *dlc*, como também teve uma correlação positiva expressiva com *price_overview*, *supported_languages* e *achievements*. Além disso, não temos problemas de multicolinariedade entre as covariáveis numéricas.

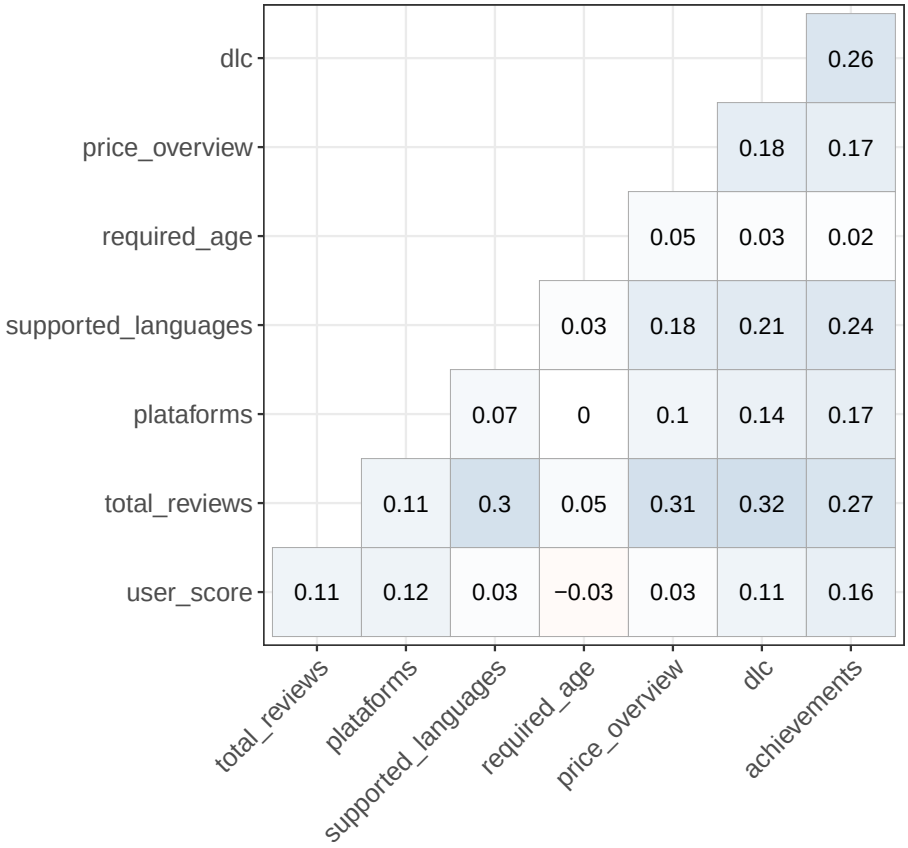


Figura 5.5: Correlação de Spearman – Variáveis Numéricas.

Fazendo a análise por aberturas de gêneros, a Figura 5.6 mostra que o comportamento da correlação entre as variáveis numéricas são bem próximos entre grupos, como também são parecidos com a visão geral de jogos. Contudo, existem algumas diferenças que podemos destacar em relação às variáveis respostas. Para os jogos que possuem os gêneros *Casual* e *Sports & Others*, existe uma correlação menor das covariáveis com a variável *total_reviews*, e para os jogos que possuem o gênero *Simulation*, existe uma correlação maior entre *user_score* e *total_reviews* em comparação aos outros gêneros de jogos.

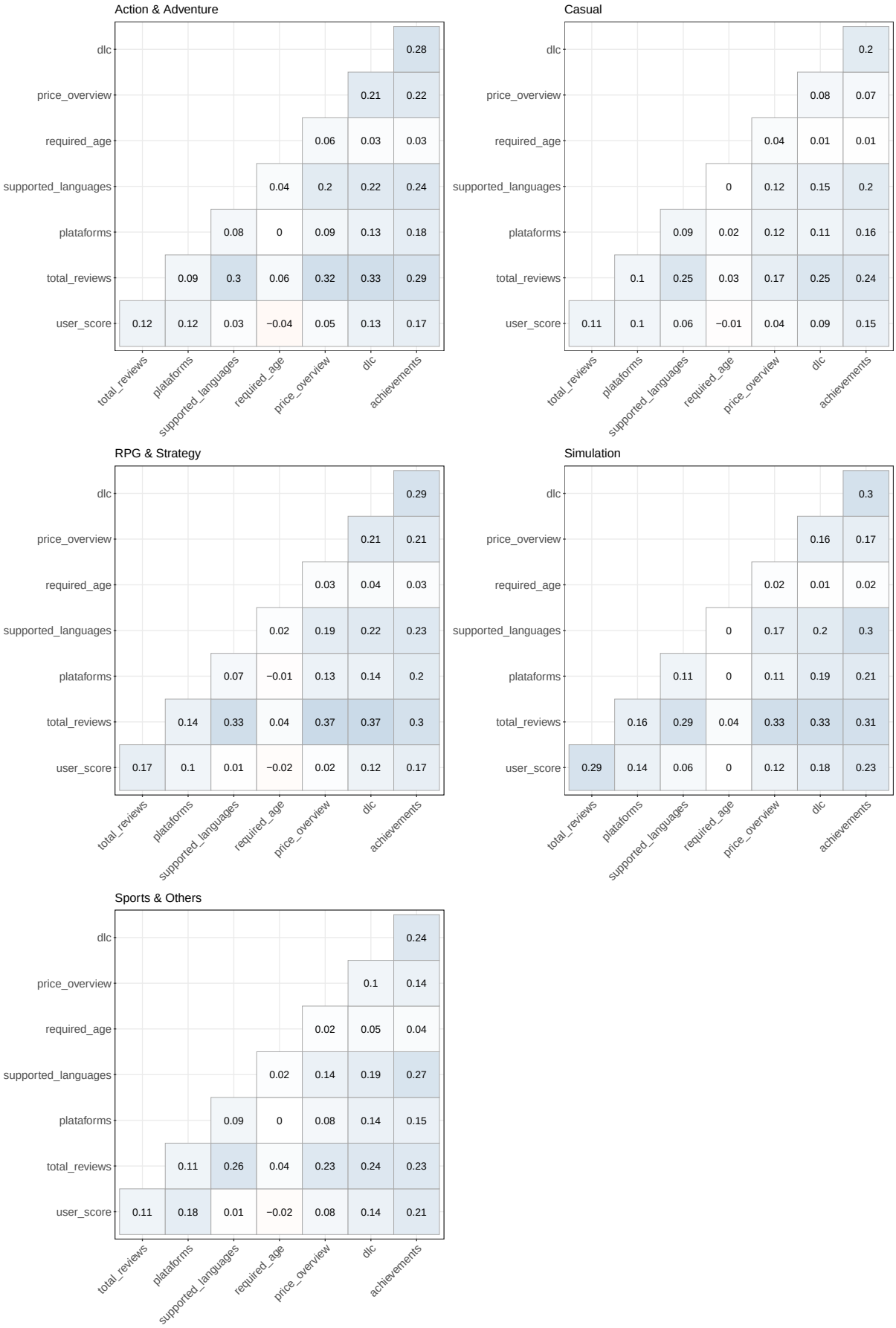


Figura 5.6: Correlação de Spearman – Abertura por Gêneros.

Jogos que possuem algum tipo de compatibilidade com o controle conseguem obter 79% dos seus jogos com classificações positivas contra 66% dos jogos que não possuem esse tipo de suporte. Além disso, jogos que utilizam *trading cards* e *workshop* possuem uma mediana na quantidade de *reviews* de pelo menos 10 vezes maior do que os jogos que não possuem essas categorias.

controller_support	**0**, N = 8,994[1]	**1**, N = 5,669[1]
new_score		
Non-Positive	3,062 (34%)	1,166 (21%)
Positive	5,932 (66%)	4,503 (79%)
[1] n (%)		

Figura 5.7: Comparação controller_support com new_score.

trading_cards	**0**, N = 13,107[1]	**1**, N = 1,556[1]
user_score	0.82 (0.67, 0.93)	0.85 (0.74, 0.92)
total_reviews	21 (10, 64)	335 (102, 1,366)
[1] Median (IQR)		

Figura 5.8: Comparação trading_cards com user_score e total_reviews.

workshop	**0**, N = 14,177[1]	**1**, N = 486[1]
user_score	0.82 (0.67, 0.92)	0.86 (0.75, 0.94)
total_reviews	24 (11, 91)	247 (30, 1,506)
[1] Median (IQR)		

Figura 5.9: Comparação workshop com user_score e total_reviews.

Ainda, conseguimos encontrar algumas diferenças e relações entre as categorias e as variáveis repostas em cada gênero, que seguem um comportamento parecido com os jogos em geral.

- Jogos gratuitos possuem uma mediana na quantidade de avaliações menor do que jogos pagos. Para os jogos com gênero Casual, aqueles que são gratuitos conseguem obter 80% dos seus jogos com classificações positivas contra 73% dos jogos pagos.

- Jogos com a opção *singleplayer* possuem uma mediana na quantidade de *reviews* menor do que jogos sem essa opção, porém a mediana do *user_score* é um pouco maior.

- Jogos com a opção *multiplayer* possuem uma mediana na quantidade de *reviews* maior do que jogos sem essa opção.

- Jogos com a opção de serem jogados *online* possuem uma mediana na quantidade de *reviews* maior do que jogos sem essa opção, porém a mediana do *user_score* é um pouco menor. Em relação ao *new_score*, jogos que não são *online* conseguem obter um percentual maior dos seus jogos com classificações positivas em comparação aos jogos *online*, com exceção daqueles com gênero Simulação, no qual o contrário ocorre.

- Jogos com algum tipo de compatibilidade com o controle (*controller_support*), *trading_cards* ou *workshop* possuem uma mediana maior na quantidade de *reviews* e no *user_score* em comparação aos jogos sem essas categorias. Além disso, conseguem obter um percentual maior de jogos com classificações positivas.

- Jogos com lançamento antecipado (*early_access*) possuem uma mediana no *user_score* menor do que jogos sem esse tipo de lançamento. Além disso, conseguem obter um percentual menor dos seus jogos com classificações positivas.

5.2 Steam Modelos

A modelagem foi realizada utilizando modelos estatísticos e de *Machine Learning* para as três variáveis respostas (*total_reviews*, *user_score* e *new_score*), tanto para a base de treino geral dos jogos *indie*, como também para cada grupo de gênero.

As técnicas utilizadas para regressão e para classificação foram:

- Modelo Linear Generalizado (GLM);
- Regressão Ridge e Lasso;
- Árvore de Decisão e *Random Forest*.

As métricas para avaliar o desempenho dos modelos para regressão serão Erro Quadrático Médio (MSE), Raiz do Erro Quadrático Médio (RMSE) e Erro Médio Absoluto (MAE), e dos modelos para classificação serão Acurácia, *Kappa*, Sensibilidade e Especificidade.

5.2.1 Modelagem Geral

total_reviews

A variável resposta *total_reviews* é uma variável numérica de contagem, portanto o primeiro modelo realizado foi um Modelo Linear Generalizado com distribuição Poisson. A Tabela 5.4 mostra as estimativas dos coeficientes das variáveis do modelo, em que todas foram significativas (p-valor = 0) e, portanto, podemos fazer algumas interpretações.

- Um aumento de 1 unidade no número de plataformas resulta em um aumento de 21.12% na quantidade de *reviews*, assumindo que todas as outras variáveis sejam fixas.
- O jogo ser gratuito resulta em uma diminuição de 67.90% na quantidade de *reviews*, assumindo que todas as outras variáveis sejam fixas.
- O jogo ter a opção de ser jogado *online* resulta em um aumento de 656.74% na quantidade de *reviews*, assumindo que todas as outras variáveis sejam fixas.
- O jogo ter a categoria *workshop* resulta em um aumento de 261.34% na quantidade de *reviews*, assumindo que todas as outras variáveis sejam fixas.
- Um aumento de 1 unidade no número de *achievements* resulta em um aumento de apenas 0.02% na quantidade de *reviews* e um aumento de 1 unidade no número de *dlc* resulta em um aumento de apenas 0.05% na quantidade de *reviews*, assumindo em ambos os casos que todas as outras variáveis sejam fixas.

Tabela 5.4: GLM – Poisson – Coeficientes – Total Reviews.

Variáveis	Estimativa	Erro Padrão	Valor z	Pr(>|z|)	Exp. Coef
(Intercept)	5.9877	0.0015	4124	0	398.4883
platforms	0.1916	0.0004	517	0	1.2112
supported_languages	0.0317	0.0000	2280	0	1.0322
required_age	0.0956	0.0001	1558	0	1.1003
is_free	-1.1362	0.0030	-380	0	0.3210
price_overview	0.0276	0.0000	2329	0	1.0280
singleplayer	-1.2686	0.0009	-1365	0	0.2812
multiplayer	-1.0724	0.0019	-559	0	0.3422
is_online	2.0238	0.0019	1043	0	7.5674
controller_support	0.6975	0.0006	1079	0	2.0087
trading_cards	1.0483	0.0007	1569	0	2.8527
workshop	1.2846	0.0007	1739	0	3.6134
Action_Adventure	0.1070	0.0007	146	0	1.1130
Casual	-0.4068	0.0007	-586	0	0.6658
RPG_Strategy	0.0232	0.0006	37	0	1.0235
Simulation	0.3199	0.0007	483	0	1.3770
Sports_Others	-0.0649	0.0008	-83	0	0.9371
early_access	0.3931	0.0008	473	0	1.4815
dlc	0.0005	0.0000	115	0	1.0005
achievements	0.0002	0.0000	149	0	1.0002

Os modelos de Regressão Ridge e Lasso também foram utilizados com uma distribuição Poisson, o parâmetro de ajuste λ foi selecionado através de validação cruzada e foi escolhido aquele que tivesse menor valor de erro médio. O parâmetro resultante para a Regressão Ridge foi de $\lambda = 212206.9$ e para o Lasso foi de $\lambda = 784.02$. A Tabela 5.5 mostra as estimativas dos coeficientes das variáveis para os modelos Ridge e Lasso, de forma que para o modelo Lasso, as covariáveis que tiveram estimativas não-nulas resultantes foram *required_age*, *price_overview*, *is_online*, *trading_cards* e *workshop*.

Tabela 5.5: Regressão Ridge e Lasso – Poisson – Coeficientes – Total Reviews.

Variáveis	Estimativa Regressão Ridge	Estimativa Lasso
(Intercept)	6.689911	4.71369
plataforms	0.001723	.
supported_languages	0.000250	.
required_age	0.002563	0.02823
is_free	-0.002826	.
price_overview	0.000664	0.01711
singleplayer	-0.022205	.
multiplayer	0.008160	.
is_online	0.014482	0.30332
controller_support	0.003193	.
trading_cards	0.014428	0.22319
workshop	0.031933	1.19341
Action_Adventure	0.001278	.
Casual	-0.002953	.
RPG_Strategy	0.001616	.
Simulation	0.001876	.
Sports_Others	0.002889	.
early_access	0.002796	.
dlc	0.000018	.
achievements	0.000001	.

Por fim, foi realizado um modelo de *Random Forest* para a variável resposta *total_reviews*. Para a seleção da melhor Floresta, foram gerados 12 modelos, variando o ajuste aleatório *mtry* de 3 a 5 e variando o tamanho do nó em 5, 50, 100 e 500. Em todas as Florestas Aleatórias geradas foi utilizado um total de 1000 árvores. A Tabela 5.6 mostra o resultado da comparação entre os modelos e a *Random Forest* selecionada foi aquela com $mtry = 3$ e $nodesize = 500$, por ter tido um menor valor de RMSE.

Tabela 5.6: Random Forest – Total Reviews.

id	mtry	nodesize	MSE	RMSE	MAE
1	3	5	40 749 364	6 383.52	**1 091.43**
2	3	50	36 595 290	6 049.40	1 115.62
3	3	100	36 421 137	6 034.99	1 145.90
4	3	500	36 305 372	**6 025.39**	1 131.26
5	4	5	46 552 795	6 822.96	1 203.73
6	4	50	38 840 111	6 232.18	1 183.94
7	4	100	36 713 718	6059.18	1 162.44
8	4	500	36 647 912	6 053.75	1 161.13
9	5	5	49 181 479	7 012.95	1 241.06
10	5	50	39 621 784	6 294.58	1 215.47
11	5	100	37 001 603	6 082.89	1 183.70
12	5	500	37 080 251	6 089.36	1 195.18

As variáveis *required_age*, *price_overview* e *supported_languages* foram as que tiveram maior nível de importância, considerando o critério de aumento na pureza do nó em cada divisão.

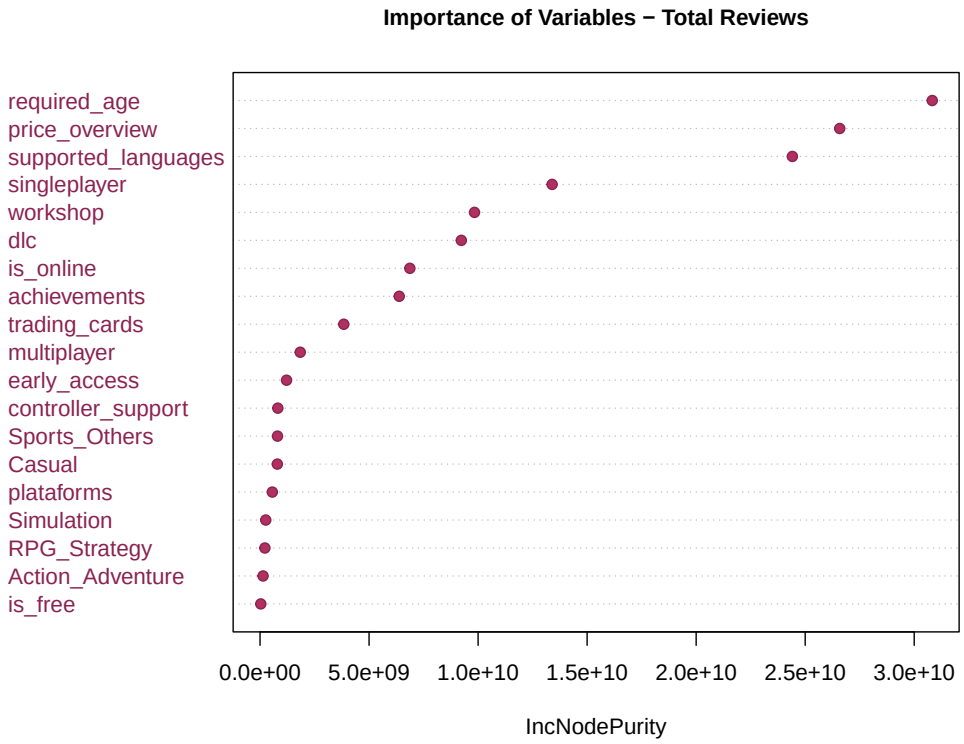


Figura 5.10: Importância das Variáveis – Random Forest – Total Reviews.

Além disso, utilizando a parametrização da melhor Floresta, foi gerada de uma Árvore de Decisão para demonstrar como seria a visualização de uma possível árvore gerada dentro da Floresta Aleatória.

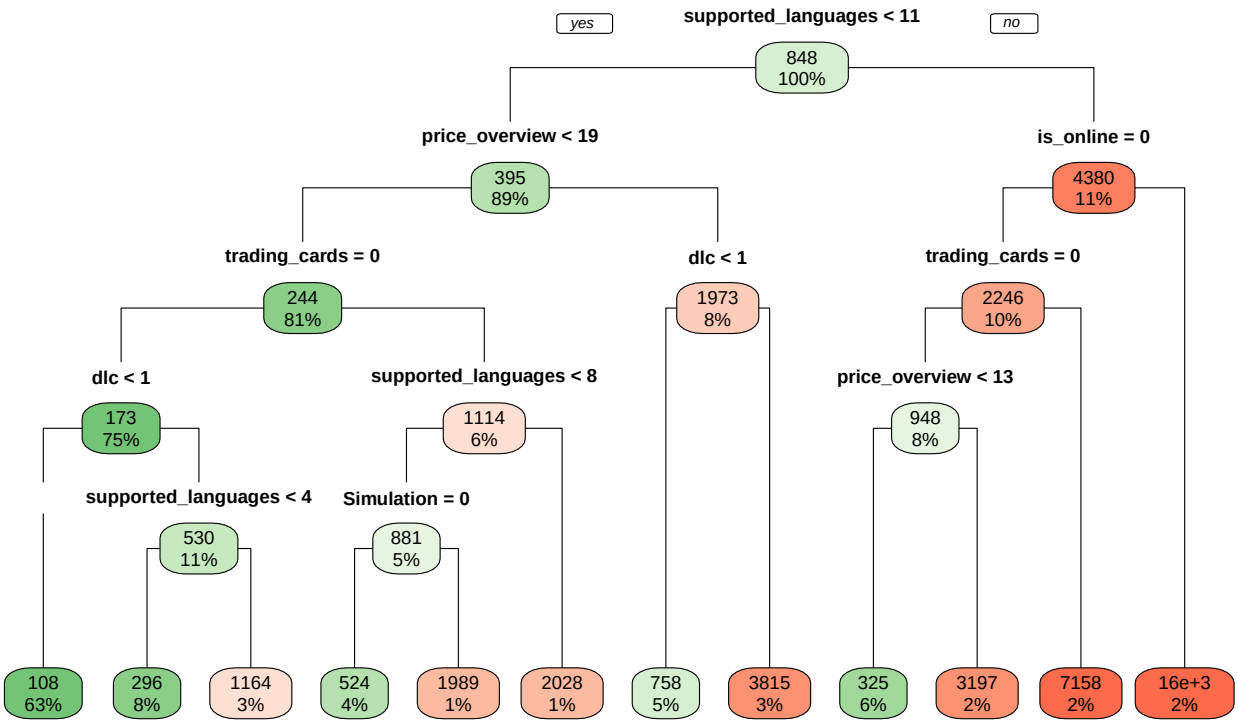


Figura 5.11: Decision Tree – Total Reviews.

A comparação feita na Tabela 5.7 mostra que a *Random Forest* foi o modelo que teve melhor desempenho em relação às métricas RMSE e MAE. Portanto, será o modelo escolhido para predizer o total de avaliações na aplicação dos jogos.

Tabela 5.7: Comparação dos Modelos – Total Reviews.

Modelo	MSE	RMSE	MAE
GLM - Poisson	1 090 539 927	33 023.32	2 119.60
Ridge Regression - Poisson	38 006 673	6 164.96	1 174.37
Lasso - Poisson	37 566 696	6 129.17	1 165.77
Random Forest	36 305 372	6 025.39	1 131.26

user_score

A variável resposta *user_score* é uma variável numérica entre 0 e 1, portanto o primeiro modelo realizado foi um Modelo Linear Generalizado com distribuição Gamma, utilizando uma correção igual 0.01 no *user_score*. A Tabela 5.8 mostra as estimativas dos coeficientes das variáveis do modelo, em que apenas 4 variáveis não foram significativas e, portanto, podemos fazer algumas interpretações.

- O jogo ser gratuito resulta em uma diminuição de 7.21% no *user_score*, assumindo que todas as outras variáveis sejam fixas.
- Um aumento de 1 unidade no preço do jogo (*price_overview*) resulta em uma diminuição de 0.19% na proporção de positivos do total de avaliações, assumindo que todas as outras variáveis sejam fixas.
- O jogo ter a opção de ser jogado *online* resulta em um aumento de 6.10% na proporção de positivos, assumindo que todas as outras variáveis sejam fixas.
- O jogo ter o gênero de Simulação resulta em um aumento de 12.34% na proporção de positivos, assumindo que todas as outras variáveis sejam fixas.
- As variáveis *singleplayer*, *multiplayer*, *dlc* e *achievements* não foram significativas.

Tabela 5.8: GLM – Gamma – Coeficientes – User Score.

Variáveis	Estimativa	Erro Padrão	Valor t	Pr(>|t|)	Exp. Coef.
(Intercept)	1.3148	0.0195	67.56	0.0000	3.7241
platforms	-0.0301	0.0037	-8.17	0.0000	0.9704
supported_languages	-0.0008	0.0002	-3.68	0.0002	0.9992
required_age	0.0090	0.0018	4.95	0.0000	1.0090
is_free	-0.0748	0.0154	-4.86	0.0000	0.9279
price_overview	-0.0019	0.0003	-5.68	0.0000	0.9981
singleplayer	-0.0101	0.0173	-0.58	0.5616	0.9900
multiplayer	-0.0102	0.0102	-0.99	0.3202	0.9899
is_online	0.0592	0.0130	4.55	0.0000	1.0610
controller_support	-0.0910	0.0058	-15.62	0.0000	0.9130
trading_cards	-0.0423	0.0086	-4.94	0.0000	0.9586
workshop	-0.0881	0.0143	-6.18	0.0000	0.9157
Action_Adventure	0.0562	0.0061	9.19	0.0000	1.0578
Casual	-0.0422	0.0056	-7.58	0.0000	0.9586
RPG_Strategy	0.0529	0.0059	9.02	0.0000	1.0543
Simulation	0.1164	0.0069	16.95	0.0000	1.1234
Sports_Others	0.0619	0.0084	7.40	0.0000	1.0639
early_access	0.0361	0.0094	3.85	0.0001	1.0367
dlc	-0.0001	0.0001	-0.69	0.4884	0.9999
achievements	0.0000	0.0000	0.42	0.6773	1.0000

Os modelos de Regressão Ridge e Lasso foram utilizados com uma distribuição Normal, o parâmetro de ajuste λ foi selecionado através de validação cruzada e foi escolhido aquele que tivesse menor valor de erro médio. O parâmetro resultante para a Regressão Ridge foi de $\lambda = 0.3955347$ e para o Lasso foi de $\lambda = 0.008289072$. A Tabela 5.9 mostra as estimativas dos coeficientes das variáveis para os modelos Ridge e Lasso, de forma que para o modelo Lasso, as covariáveis que tiveram estimativas não-nulas resultantes foram *platforms*, *supported_languages*, *is_online*, *controller_support*, *trading_cards*, *workshop*, *Action_Adventure*, *Casual*, *RPG_Strategy*, *Simulation*, *Sports_Others* e *early_access*.

Por fim, foi realizado um modelo de *Random Forest* para a variável resposta *user_score*. Para a seleção da melhor Floresta, foram gerados 12 modelos, variando o ajuste aleatório *mtry* de 3 a 5 e variando o tamanho do nó em 5, 50, 100 e 500. Em todas as Florestas Aleatórias geradas foi utilizado um total de 1000 árvores. A Tabela 5.10 mostra o resultado da comparação entre os modelos e a *Random Forest* selecionada foi aquela com $mtry = 3$ e $nodesize = 100$, por ter tido um menor valor de RMSE.

Tabela 5.9: Regressão Ridge e Lasso – Normal – Coeficientes – User Score.

Variáveis	Estimativa Regressão Ridge	Estimativa Lasso
(Intercept)	0.765858	0.77581
platforms	0.008547	0.01265
supported_languages	0.000259	0.00003
required_age	-0.001512	.
is_free	0.011290	.
price_overview	0.000312	.
singleplayer	0.007464	.
multiplayer	0.000067	.
is_online	-0.010446	-0.00522
controller_support	0.020492	0.04190
trading_cards	0.013022	0.01037
workshop	0.017400	0.01010
Action_Adventure	-0.005679	-0.00813
Casual	0.007295	0.00939
RPG_Strategy	-0.012164	-0.01335
Simulation	-0.023517	-0.04703
Sports_Others	-0.011057	-0.00922
early_access	-0.013295	-0.00661
dlc	0.000024	.
achievements	0.000001	.

Tabela 5.10: Random Forest – User Score.

id	mtry	nodesize	MSE	RMSE	MAE
1	3	5	0.03104	0.17619	0.14381
2	3	50	0.03086	0.17566	0.14364
3	3	100	0.03079	**0.17547**	0.14362
4	3	500	0.03100	0.17607	0.14474
5	4	5	0.03145	0.17733	0.14394
6	4	50	0.03095	0.17593	**0.14336**
7	4	100	0.03088	0.17573	0.14345
8	4	500	0.03098	0.17602	0.14446
9	5	5	0.03194	0.17870	0.14451
10	5	50	0.03110	0.17636	0.14348
11	5	100	0.03098	0.17602	0.14352
12	5	500	0.03103	0.17616	0.14445

As variáveis *achievements*, *price_overview* e *Simulation* foram as que tiveram maior nível de importância, considerando o critério de aumento na pureza do nó em cada divisão.

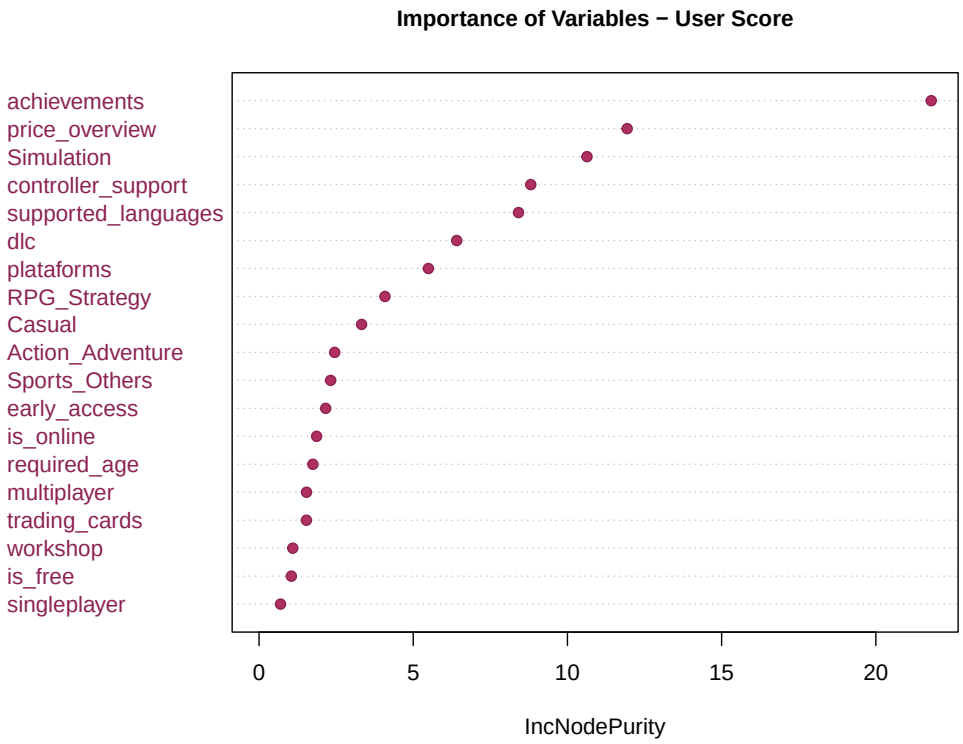


Figura 5.12: Importância das Variáveis – Random Forest – User Score.

Além disso, utilizando a parametrização da melhor Floresta, foi gerada uma Árvore de Decisão para demonstrar como seria a visualização de uma possível árvore gerada dentro da Floresta Aleatória.

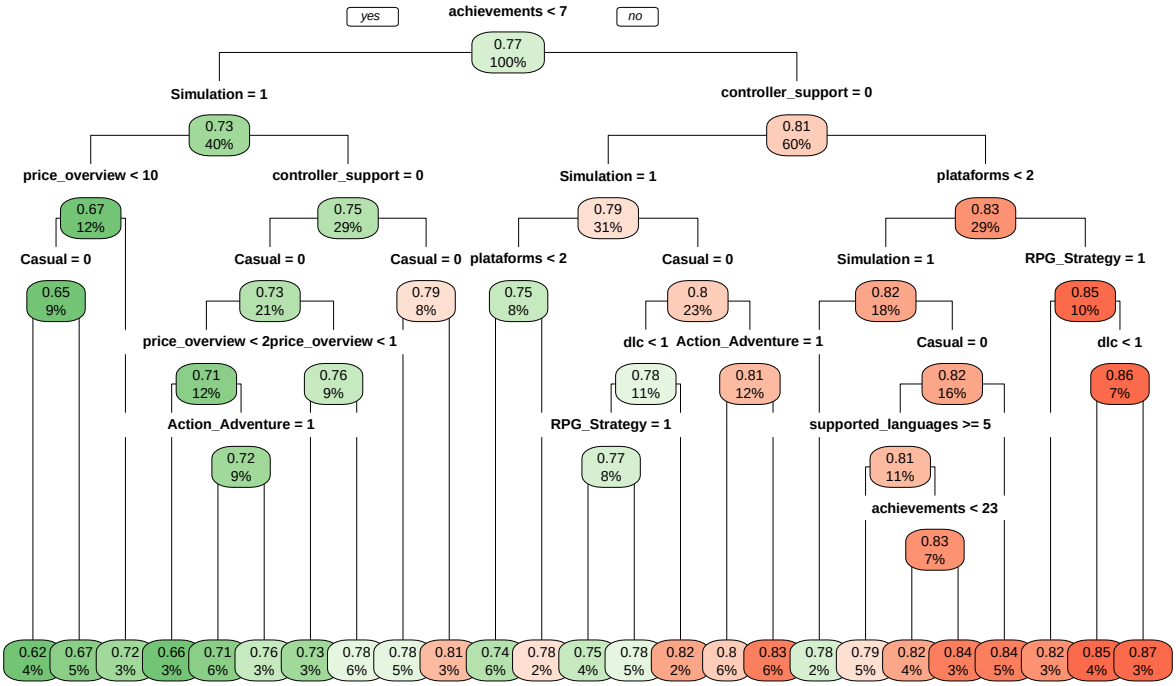


Figura 5.13: Decision Tree – User Score.

A comparação feita na Tabela 5.11 mostra que a *Random Forest* foi o modelo que teve melhor desempenho em relação às métricas RMSE e MAE. Portanto, será o modelo escolhido para predizer a proporção de positivos na aplicação dos jogos.

Tabela 5.11: Comparação dos Modelos – User Score.

Modelo	MSE	RMSE	MAE
GLM - Gamma	0.0319	0.1786	0.1467
Ridge Regression - Normal	0.0322	0.1793	0.1485
Lasso - Normal	0.0319	0.1787	0.1481
Random Forest	0.0308	0.1755	0.1436

new_score

A variável resposta *new_score* é uma variável categórica binária, portanto o primeiro modelo realizado foi um Modelo Linear Generalizado com distribuição Binomial. A Tabela 5.12 mostra as estimativas dos coeficientes das variáveis do modelo, em que apenas 3 variáveis não foram significativas e, portanto, podemos fazer algumas interpretações.

- Um aumento de 1 unidade na quantidade de idiomas disponíveis resulta em um aumento de 1.20% na probabilidade do *new_score* ser Positivo, assumindo que todas as outras variáveis sejam fixas.
- O jogo ser gratuito resulta em um aumento de 35.74% na probabilidade do *new_score* ser Positivo, assumindo que todas as outras variáveis sejam fixas.
- O jogo ter a opção de ser jogado *online* resulta em uma diminuição de 35.44% na probabilidade do *new_score* ser Positivo, assumindo que todas as outras variáveis sejam fixas.
- O jogo ter *workshop* resulta em um aumento de 89.90% na probabilidade do *new_score* ser Positivo, assumindo que todas as outras variáveis sejam fixas.
- O jogo ter o gênero Casual resulta em um aumento de 35.13% na proporção de positivos, assumindo que todas as outras variáveis sejam fixas.
- O jogo ter o gênero Simulação resulta em uma diminuição de 47.42% na proporção de positivos, assumindo que todas as outras variáveis sejam fixas.
- As variáveis *singleplayer*, *multiplayer* e *achievements* não foram significativas.

Tabela 5.12: GLM – Binomial – Coeficientes – New Score.

Variáveis	Estimativa	Erro Padrão	Valor z	Pr(>|z|)	Exp. Coef.
(Intercept)	0.5246	0.1346	3.90	0.0001	1.6898
platforms	0.2107	0.0296	7.12	0.0000	1.2346
supported_languages	0.0119	0.0022	5.43	0.0000	1.0120
required_age	-0.0444	0.0115	-3.87	0.0001	0.9565
is_free	0.3056	0.1205	2.54	0.0112	1.3574
price_overview	0.0177	0.0031	5.74	0.0000	1.0179
singleplayer	0.0536	0.1179	0.45	0.6495	1.0550
multiplayer	0.0798	0.0846	0.94	0.3454	1.0831
is_online	-0.4376	0.1002	-4.37	0.0000	0.6456
controller_support	0.5619	0.0440	12.76	0.0000	1.7540
trading_cards	0.4025	0.0748	5.38	0.0000	1.4956
workshop	0.6413	0.1272	5.04	0.0000	1.8990
Action_Adventure	-0.2959	0.0437	-6.77	0.0000	0.7439
Casual	0.3010	0.0404	7.46	0.0000	1.3513
RPG_Strategy	-0.3062	0.0409	-7.48	0.0000	0.7362
Simulation	-0.6428	0.0449	-14.33	0.0000	0.5258
Sports_Others	-0.2895	0.0567	-5.10	0.0000	0.7486
early_access	-0.1613	0.0613	-2.63	0.0085	0.8510
dlc	0.1288	0.0233	5.53	0.0000	1.1375
achievements	0.0000	0.0001	0.01	0.9893	1.0000

Os modelos de Regressão Ridge e Lasso foram utilizados com uma distribuição Binomial, o parâmetro de ajuste λ foi selecionado através de validação cruzada e foi escolhido aquele que tivesse menor valor de erro médio. O parâmetro resultante para a Regressão Ridge foi de $\lambda = 0.0006562924$ e para o Lasso foi de $\lambda = 0.002165788$. A Tabela 5.13 mostra as estimativas dos coeficientes das variáveis para os modelos Ridge e Lasso, de forma que para o modelo Lasso, as únicas covariáveis que tiveram estimativas nulas foram *multiplayer* e *achievements*.

Tabela 5.13: Regressão Ridge e Lasso – Binomial – Coeficientes – New Score.

Variáveis	Estimativa Regressão Ridge	Estimativa Lasso
(Intercept)	0.54097	0.57723
platforms	0.21498	0.20679
supported_languages	0.01217	0.01121
required_age	-0.04283	-0.03467
is_free	0.36268	0.30386
price_overview	0.01898	0.01730
singleplayer	0.05438	0.00737
multiplayer	0.07683	.
is_online	-0.42213	-0.31343
controller_support	0.56979	0.56181
trading_cards	0.44442	0.43999
workshop	0.65504	0.59316
Action_Adventure	-0.29419	-0.26172
Casual	0.30145	0.27640
RPG_Strategy	-0.29920	-0.26897
Simulation	-0.63624	-0.60524
Sports_Others	-0.29242	-0.25859
early_access	-0.17890	-0.17159
dlc	0.04591	0.00053
achievements	0.00001	.

Por fim, foi realizado um modelo de *Random Forest* para a variável resposta *new_score*. Para a seleção da melhor Floresta, foram gerados 12 modelos, variando o ajuste aleatório *mtry* de 3 a 5 e variando o tamanho do nó em 5, 50, 100 e 500. Em todas as Florestas Aleatórias geradas foi utilizado um total de 1000 árvores. A Tabela 5.14 mostra o resultado da comparação entre os modelos e a *Random Forest* selecionada foi aquela com $mtry = 3$ e $nodesize = 500$, por ter tido um maior valor de acurácia e sensibilidade.

Tabela 5.14: Random Forest – New Score.

id	mtry	nodesize	Acurácia	Kappa	Sensibilidade	Especificidade
1	3	5	0.8135	0.1753	0.5019	0.8344
2	3	50	0.8144	0.1375	0.5135	0.8287
3	3	100	0.8137	0.1260	0.5058	0.8272
4	3	500	**0.8166**	0.0817	**0.5833**	0.8215
5	4	5	0.8049	0.1865	0.4509	0.8377
6	4	50	0.8110	0.1690	0.4830	0.8338
7	4	100	0.8130	0.1501	0.4977	0.8307
8	4	500	0.8149	0.0974	0.5294	0.8235
9	5	5	0.7966	**0.1911**	0.4194	**0.8400**
10	5	50	0.8086	0.1793	0.4686	0.8358
11	5	100	0.8103	0.1515	0.4750	0.8312
12	5	500	0.8140	0.1125	0.5101	0.8255

Porém, como os dados são desbalanceados, a Floresta Aleatória não consegue detectar direito a classe *Non-Positive* conforme a Figura 5.14. Portanto, estaremos ajustando a Floresta final com pesos $w_1 = 0.4$ e $w_2 = 0.6$ para as classes *Non-Positive* e *Positive*, respectivamente. A Figura 5.15 ilustra a matriz de confusão do modelo final com os pesos, de forma que conseguimos detectar melhor a classe de *Non-Positive*.

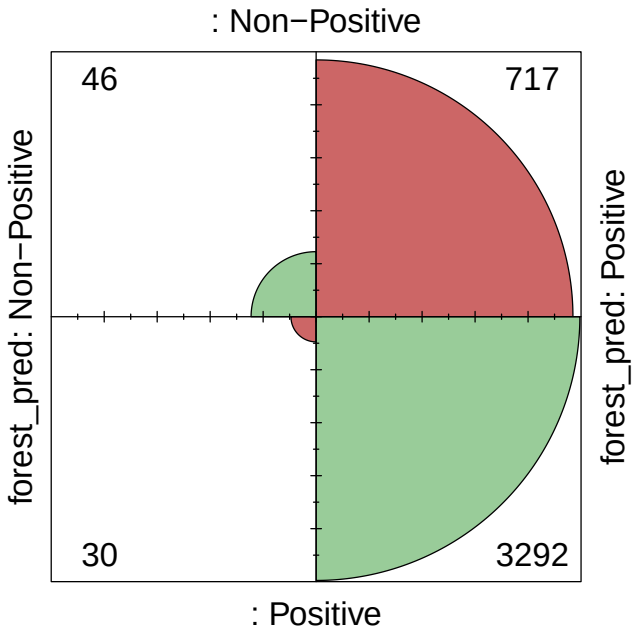


Figura 5.14: Matriz de Confusão – Random Forest – New Score (mtry = 3, nodesize = 500).

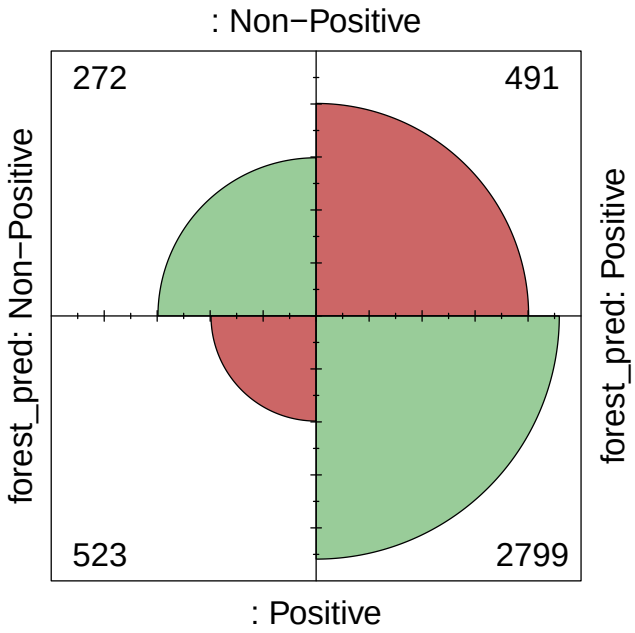


Figura 5.15: Matriz de Confusão – Random Forest – New Score (mtry = 3, nodesize = 500, $w_1 = 0.4$ e $w_2 = 0.6$).

As variáveis *achievements*, *Simulation* e *controller_support* foram as que tiveram maior nível de importância, considerando o critério do Índice de Gini.

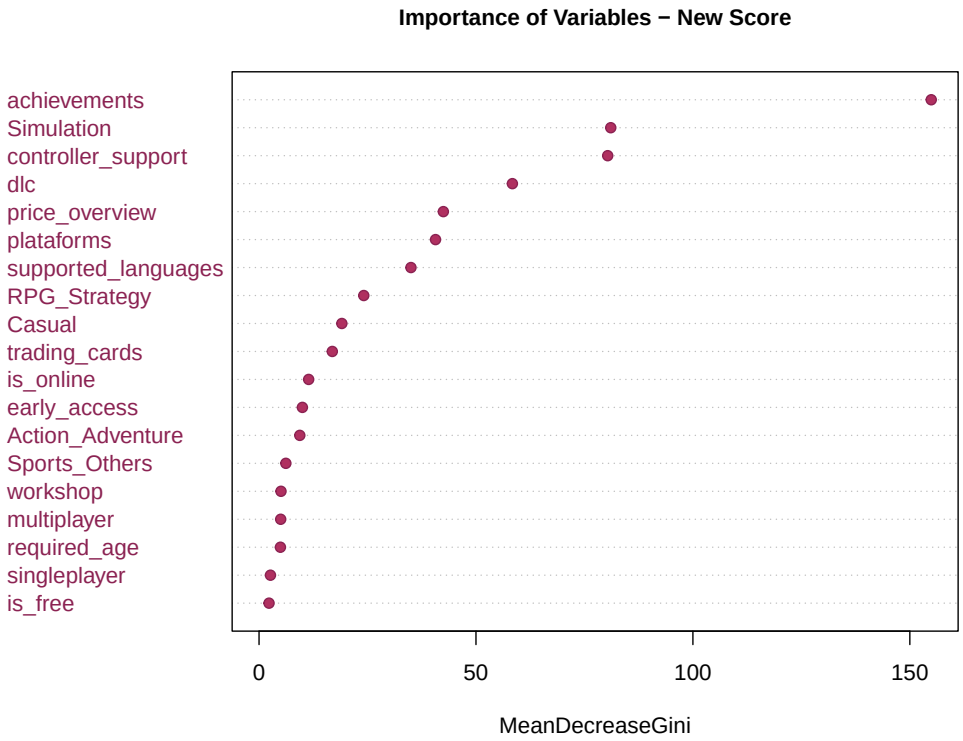


Figura 5.16: Importância das Variáveis – Random Forest – New Score.

Além disso, utilizando a parametrização da melhor Floresta com pesos, foi gerada uma Árvore de Decisão com pesos para demonstrar como seria a visualização de uma possível árvore gerada dentro da Floresta Aleatória.

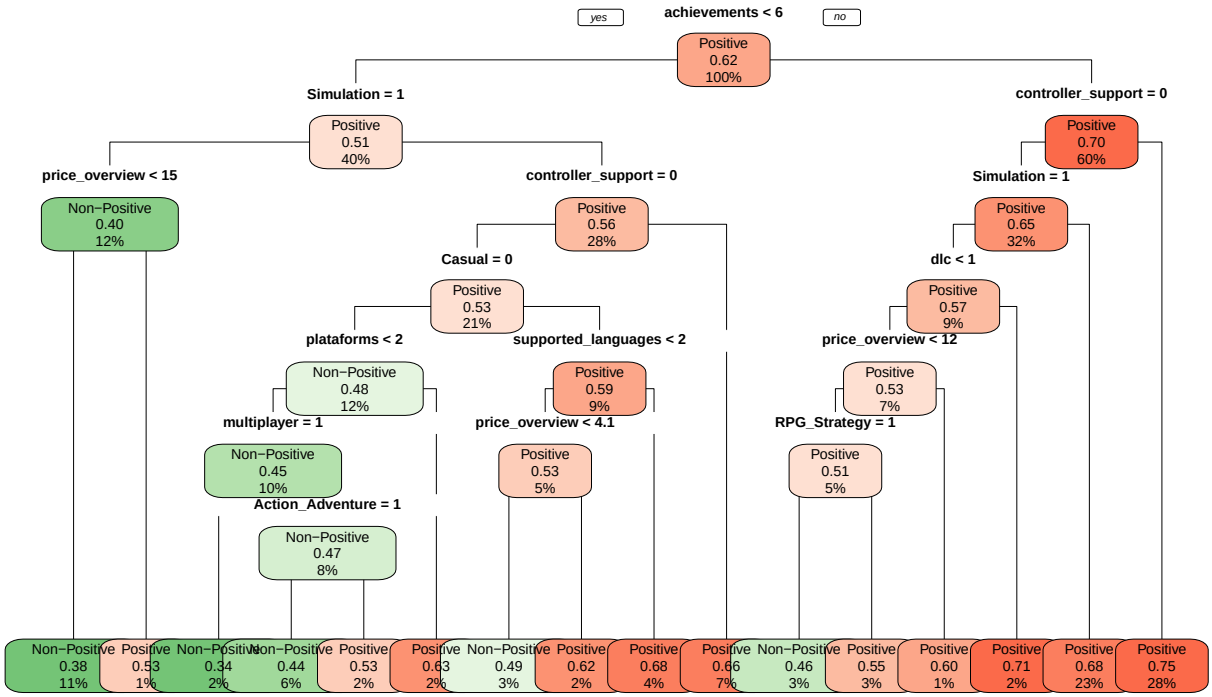


Figura 5.17: Decision Tree – New Score.

A comparação feita na Tabela 5.15 mostra que a *Random Forest* foi o modelo que teve melhor desempenho em relação às métricas acurácia e *kappa*. Portanto, será o modelo escolhido para predizer o *new_score* na aplicação dos jogos.

Tabela 5.15: Comparação dos Modelos – New Score.

Modelo	Acurácia	Kappa	Sensibilidade	Especificidade
GLM - Binomial	0.6002	0.1425	0.2613	0.8732
Ridge Regression - Binomial	0.5993	0.1359	0.2580	0.8697
Lasso - Binomial	0.6007	0.1384	0.2594	0.8707
Random Forest (Pesos)	0.7518	0.1959	0.3421	0.8508

5.2.2 Modelagem por Gênero dos Jogos

Seguindo a metodologia realizada para a base geral dos jogos *indie*, foram feitas as mesmas modelagens para cada grupo de gênero. Porém, devido ao volume de modelos, variáveis respostas e informações, nessa seção apenas constará alguns resultados das modelagens realizadas para a variável resposta *total_reviews*, com foco nos gráficos relacionados a *Random Forest* e nas tabelas resumo de comparação dos modelos.

Action & Adventure

Os mesmos modelos utilizados na base geral de jogos *indie* também foram usados para a segmentação da base com os jogos que possuem o gênero de Ação e Aventura. Portanto, para a variável resposta numérica de contagem *total_reviews*, o modelo de *Random Forest* selecionado foi aquele com $mtry = 3$ e $nodesize = 100$, por ter tido um menor valor de RMSE.

As variáveis *required_age*, *supported_languages* e *price_overview* foram as que tiveram maior nível de importância, considerando o critério de aumento na pureza do nó em cada divisão.

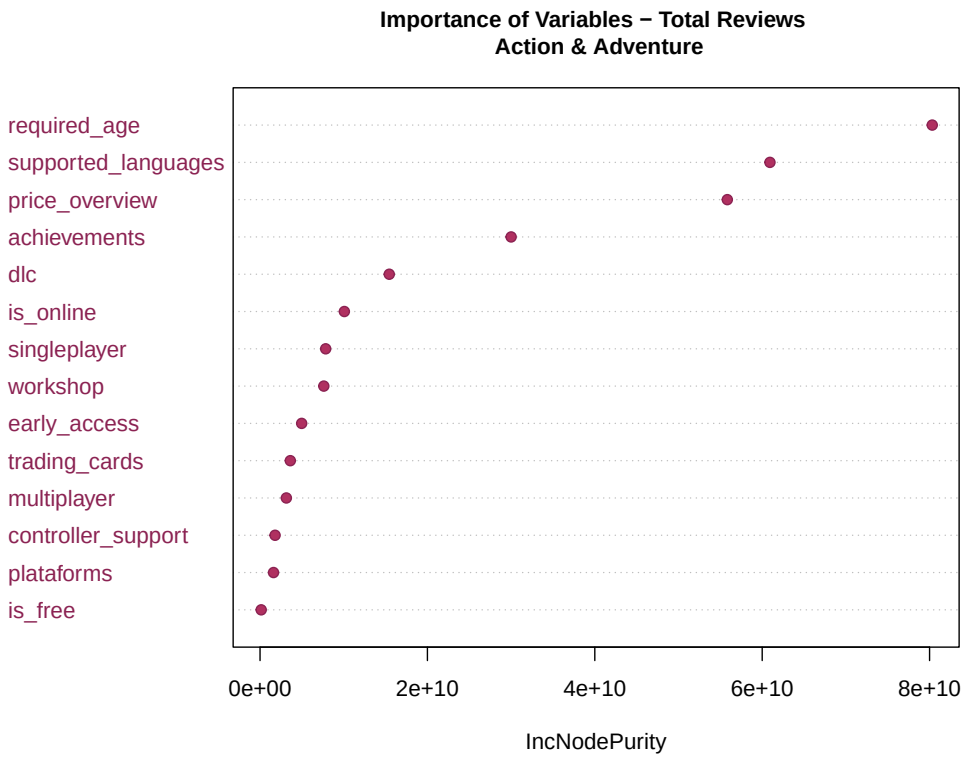


Figura 5.18: Importância das Variáveis – Random Forest – Total Reviews Action & Adventure.

Além disso, utilizando a parametrização da melhor Floresta, foi gerada de uma Árvore de Decisão para demonstrar como seria a visualização de uma possível árvore gerada dentro da Floresta Aleatória.

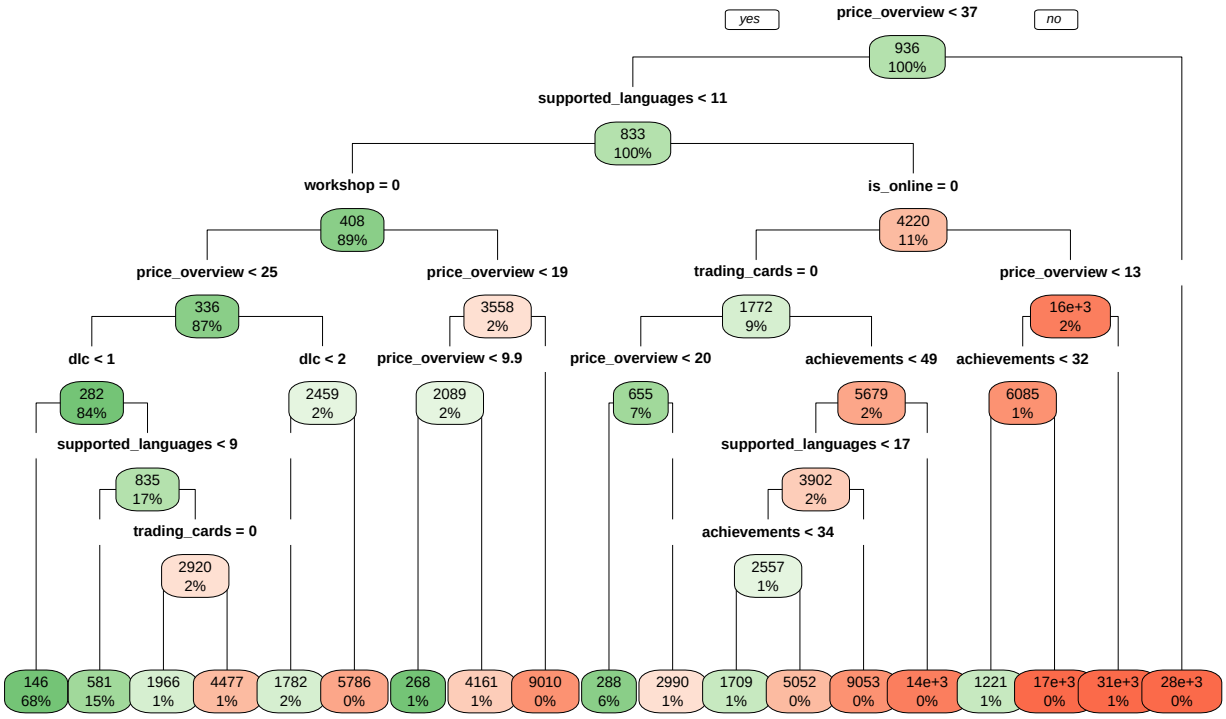


Figura 5.19: Decision Tree – Total Reviews – Action & Adventure.

A comparação feita na Tabela 5.16 mostra que a *Random Forest* foi o modelo que teve melhor desempenho em relação às métricas. Portanto, será o modelo escolhido para predizer o total de avaliações na aplicação dos jogos com o gênero Ação e Aventura.

Tabela 5.16: Comparação dos Modelos – Total Reviews – Action & Adventure.

Modelo	MSE	RMSE	MAE
GLM - Poisson	7 453 306 928	86 332.54	4 602.72
Ridge Regression - Poisson	714 717 300	26 734.20	2 251.71
Lasso - Poisson	4 849 375 408	69 637.46	3 903.61
Random Forest	52 564 200	7 250.12	1 370.34

Casual

Os mesmos modelos utilizados na base geral de jogos *indie* também foram usados para a segmentação da base com os jogos que possuem o gênero Casual. Portanto, para a variável resposta numérica de contagem *total_reviews*, o modelo de *Random Forest* selecionado foi aquele com $mtry = 3$ e $nodesize = 500$, por ter tido um menor valor de RMSE.

As variáveis *singleplayer*, *price_overview* e *workshop* foram as que tiveram maior nível de importância, considerando o critério de aumento na pureza do nó em cada divisão.

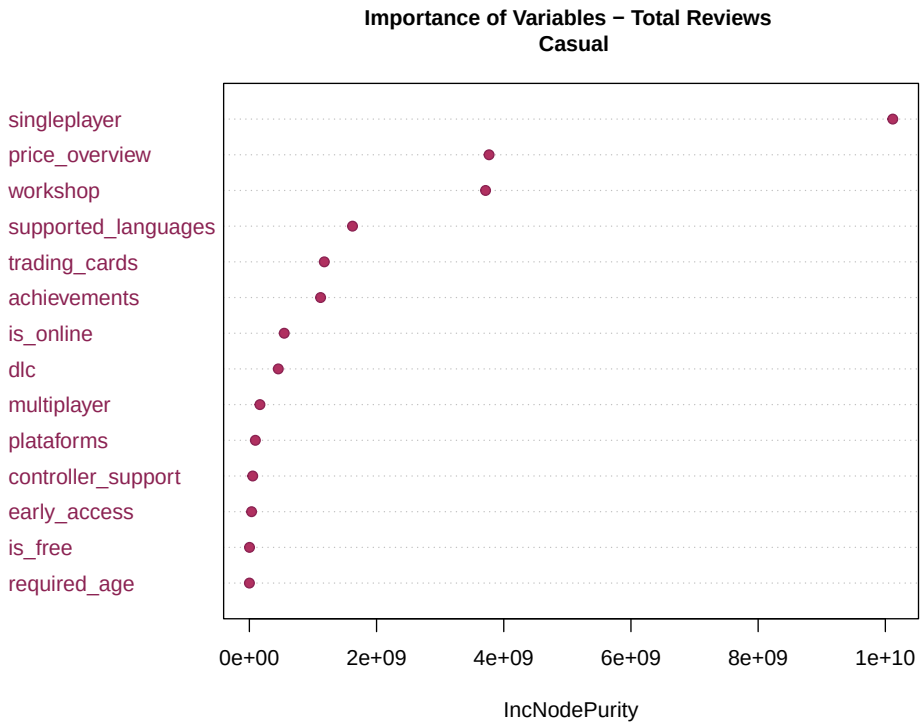


Figura 5.20: Importância das Variáveis – Random Forest – Total Reviews – Casual.

Além disso, utilizando a parametrização da melhor Floresta, foi gerada de uma Árvore de Decisão para demonstrar como seria a visualização de uma possível árvore gerada dentro da Floresta Aleatória.

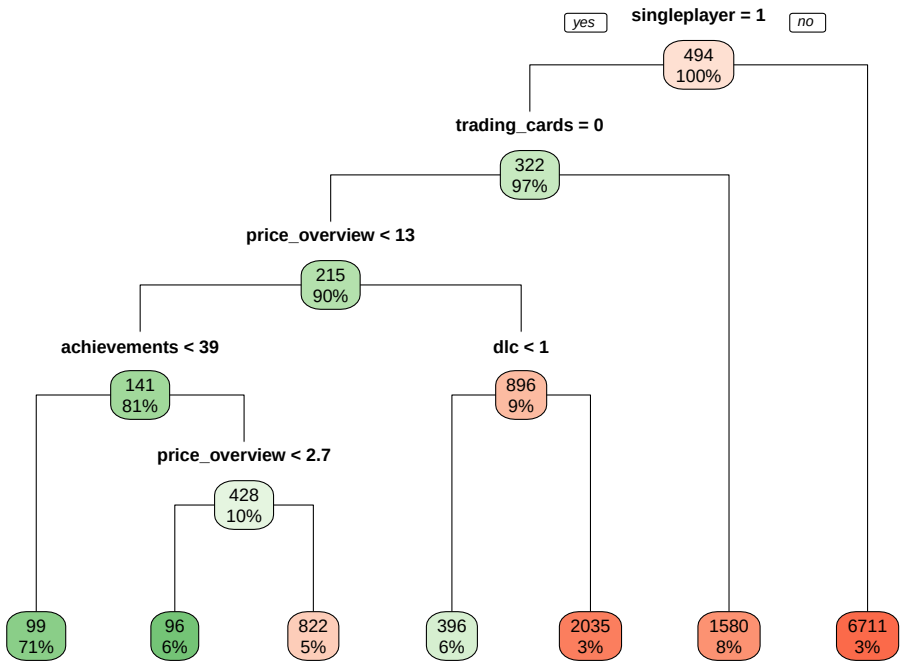


Figura 5.21: Decision Tree – Total Reviews – Casual.

A comparação feita na Tabela 5.17 mostra que a *Random Forest* foi o modelo que teve melhor desempenho em relação à métrica RMSE. Portanto, será o modelo escolhido para predizer o total de avaliações na aplicação dos jogos com o gênero Casual.

Tabela 5.17: Comparação dos Modelos – Total Reviews – Casual.

Modelo	MSE	RMSE	MAE
GLM - Poisson	35 510 893	5 959.10	838.28
Ridge Regression - Poisson	21 468 152	4 633.37	689.86
Lasso - Poisson	21 423 939	4 628.60	624.17
Random Forest	21 281 758	4 613.22	741.79

RPG & Strategy

Os mesmos modelos utilizados na base geral de jogos *indie* também foram usados para a segmentação da base com os jogos que possuem o gênero de RPG e Estratégia. Portanto, para a variável resposta numérica de contagem *total_reviews*, o modelo de *Random Forest* selecionado foi aquele com $mtry = 3$ e $nodesize = 50$, por ter tido um menor valor de RMSE.

As variáveis *price_overview*, *supported_languages* e *required_age* foram as que tiveram maior nível de importância, considerando o critério de aumento na pureza do nó em cada divisão.

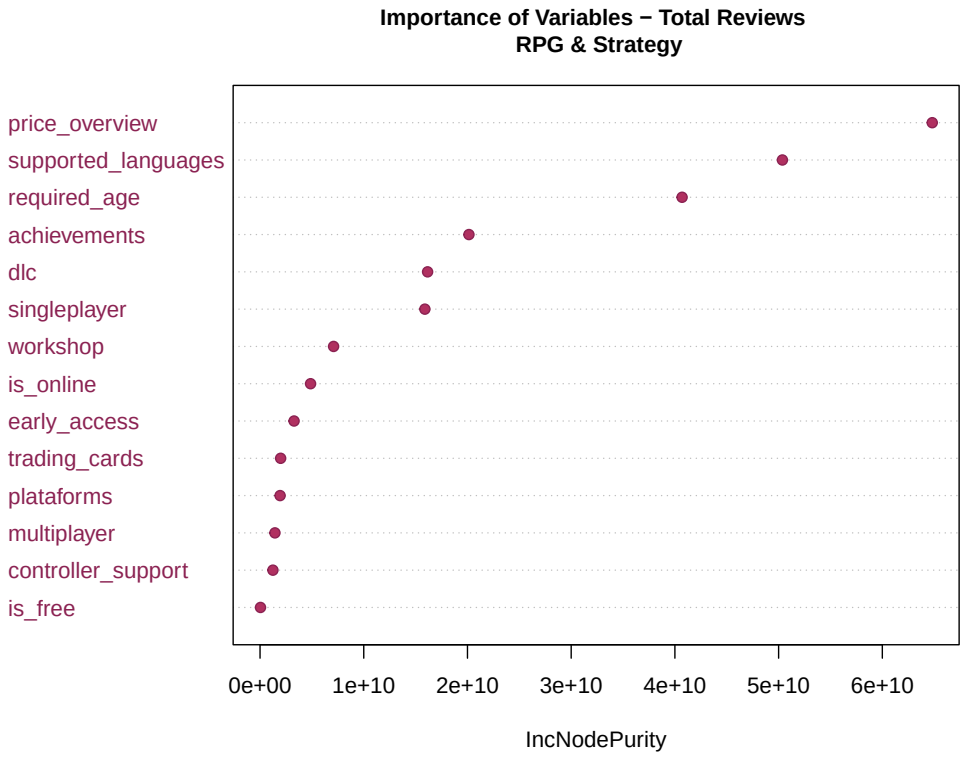


Figura 5.22: Importância das Variáveis – Random Forest – Total Reviews RPG & Strategy.

Além disso, utilizando a parametrização da melhor Floresta, foi gerada de uma Árvore de Decisão para demonstrar como seria a visualização de uma possível árvore gerada dentro da Floresta Aleatória.

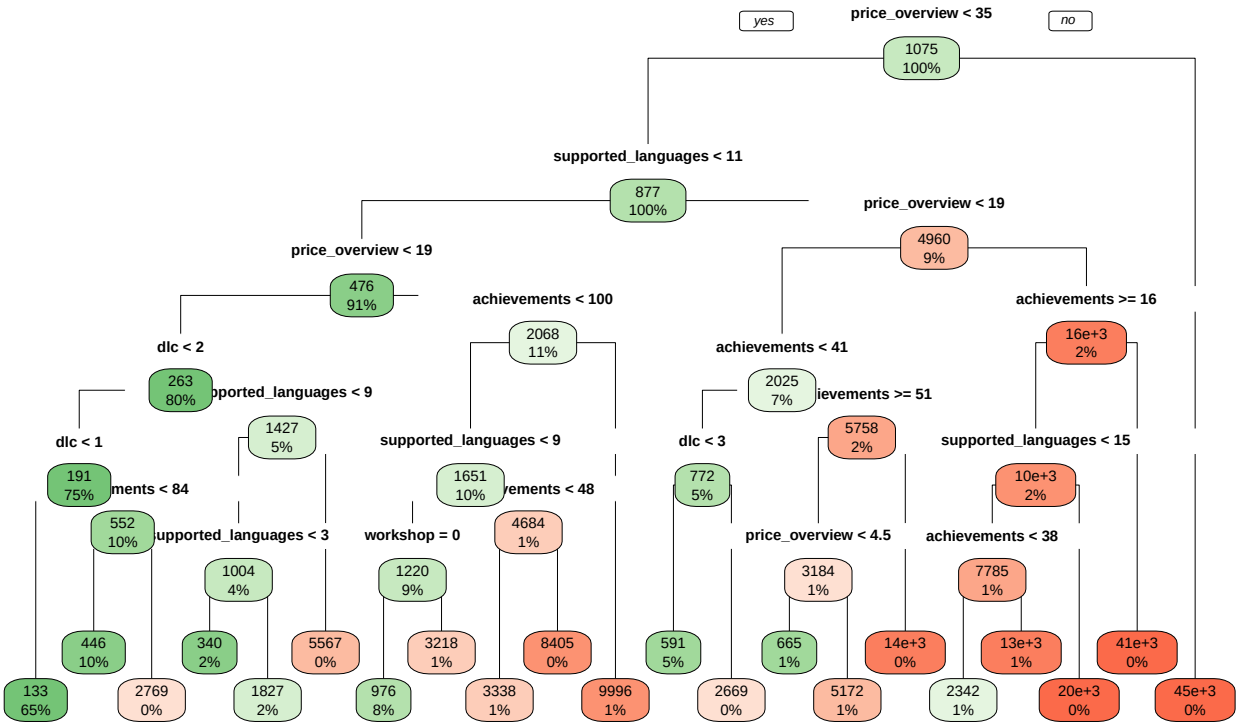


Figura 5.23: Decision Tree – Total Reviews – RPG & Strategy.

A comparação feita na Tabela 5.18 mostra que a *Random Forest* foi o modelo que teve melhor desempenho em relação à métrica RMSE. Portanto, será o modelo escolhido para predizer o total de avaliações na aplicação dos jogos com o gênero RPG e Estratégia.

Tabela 5.18: Comparação dos Modelos – Total Reviews – RPG & Strategy.

Modelo	MSE	RMSE	MAE
GLM - Poisson	74 035 304 720	272 094.30	12 380.36
Ridge Regression - Poisson	56 416 775	7 511.11	1 568.80
Lasso - Poisson	211 839 535	14 554.71	2 102.27
Random Forest	54 347 287	7 372.06	1 646.11

Simulation

Os mesmos modelos utilizados na base geral de jogos *indie* também foram usados para a segmentação da base com os jogos que possuem o gênero de Simulação. Portanto, para a variável resposta numérica de contagem *total_reviews*, o modelo de *Random Forest* selecionado foi aquele com $mtry = 3$ e $nodesize = 500$, por ter tido um menor valor de RMSE.

A variável *required_age* foi a que teve maior nível de importância, considerando o critério de aumento na pureza do nó em cada divisão.

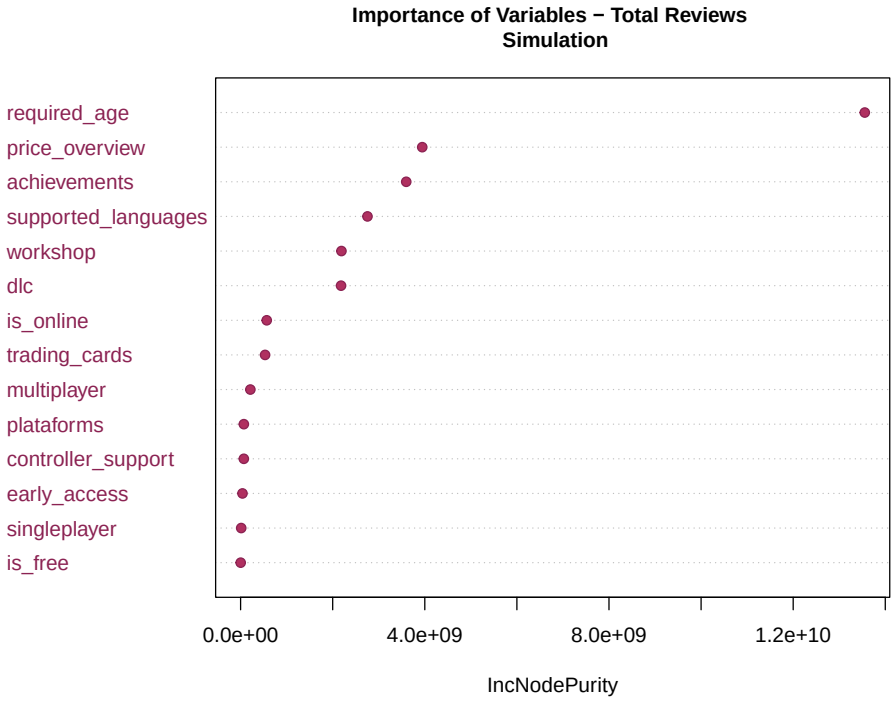


Figura 5.24: Importância das Variáveis – Random Forest – Total Reviews Simulation.

Além disso, utilizando a parametrização da melhor Floresta, foi gerada de uma Árvore de Decisão para demonstrar como seria a visualização de uma possível árvore gerada dentro da Floresta Aleatória.

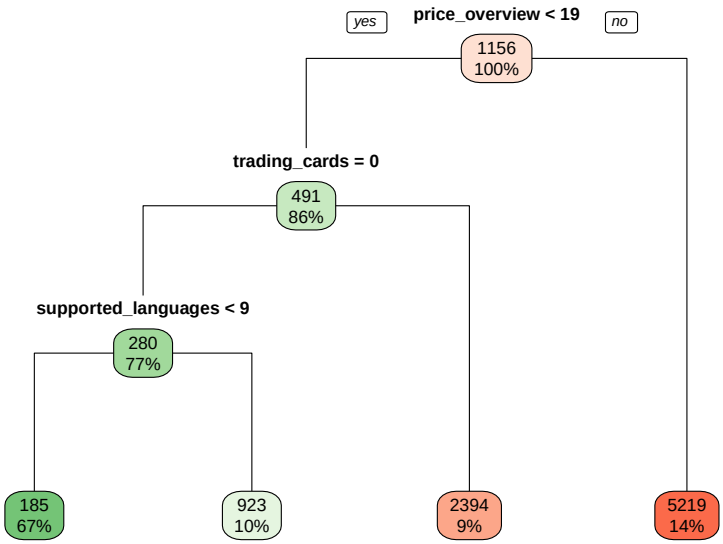


Figura 5.25: Decision Tree – Total Reviews – Simulation.

A comparação feita na Tabela 5.19 mostra que a *Random Forest* foi o modelo que teve melhor desempenho em relação à métrica RMSE. Portanto, será o modelo escolhido para predizer o total de avaliações na aplicação dos jogos com o gênero Simulação.

Tabela 5.19: Comparação dos Modelos – Total Reviews – Simulation.

Modelo	MSE	RMSE	MAE
GLM - Poisson	8 153 771 565	90 298.24	7 219.46
Ridge Regression - Poisson	121 561 560	11 025.50	2 052.09
Lasso - Poisson	82 530 186	9 084.61	1 676.75
Random Forest	80 638 947	8 979.92	1 714.63

Sports & Others

Os mesmos modelos utilizados na base geral de jogos *indie* também foram usados para a segmentação da base com os jogos que possuem o gênero Esportes e Outros. Portanto, para a variável resposta numérica de contagem *total_reviews*, o modelo de *Random Forest* selecionado foi aquele com $mtry = 3$ e $nodesize = 500$, por ter tido um menor valor de RMSE.

As variáveis *required_age* e *price_overview* foram as que tiveram maior nível de importância, considerando o critério de aumento na pureza do nó em cada divisão.

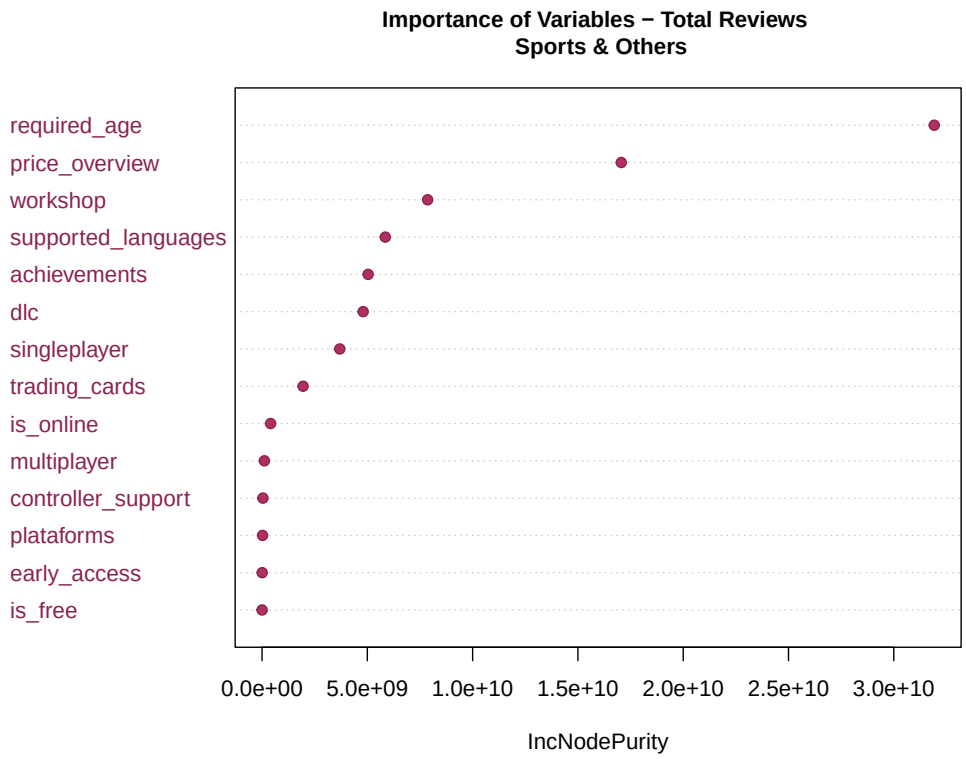


Figura 5.26: Importância das Variáveis – Random Forest – Total Reviews Sports & Others.

Além disso, utilizando a parametrização da melhor Floresta, foi gerada de uma Árvore de Decisão para demonstrar como seria a visualização de uma possível árvore gerada dentro da Floresta Aleatória.

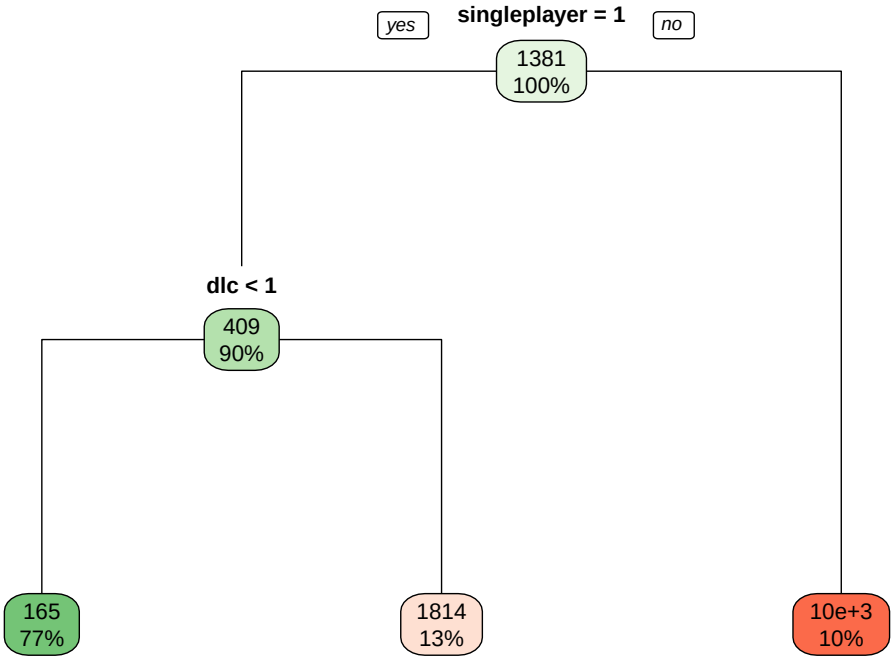


Figura 5.27: Decision Tree – Total Reviews – Sports & Others.

A comparação feita na Tabela 5.20 mostra que a Regressão Ridge foi o modelo que teve melhor desempenho em relação à métrica RMSE. Portanto, será o modelo escolhido para predizer o total de avaliações na aplicação dos jogos com o gênero Esportes e Outros.

Tabela 5.20: Comparação dos Modelos – Total Reviews – Sports & Others.

Modelo	MSE	RMSE	MAE
GLM - Poisson	1 215 214 083	34 859.92	2 400.14
Ridge Regression - Poisson	8 954 962	2 992.48	1 266.88
Lasso - Poisson	26 530 153	5 150.74	1 039.47
Random Forest	19 797 926	4 449.49	2 302.04

5.3 Aplicações do Modelo

Nessa seção serão aplicados os modelos selecionados em jogos escolhidos aleatoriamente da base de jogos *indie* com pelo menos 5 avaliações lançados em 2023. No total serão sorteados 6 jogos, sendo que um dos jogos representará os *indie* num geral e os outros 5 corresponderão a cada grupo de gênero. Por fim, serão comparadas as variáveis *total_reviews*, *user_score* e *new_score* geradas pelos modelos com as informações disponíveis na base de dados da Steam.

Geral

O jogo sorteado para a base geral dos jogos *indie* foi o *Spider Queen cave* (app_id = 1197180), lançado em 04 de Fevereiro de 2023. É um jogo pago no valor de 9.99 dólares, possui 8 diferentes idiomas disponíveis, conta com suporte a algum tipo de controle, porém não possui *workshop* e *trading cards*. Além disso, é *singleplayer*, não tem opção de ser jogado online, possui um total de 16 *achievements*, não possui *dlc* e o jogo contém os grupos de gêneros Ação e Aventura, Casual e RPG e Estratégia.

Os modelos previram que o *Spider Queen cave* teria 245 *reviews*, um *user_score* igual a 0.7805 e uma classificação positiva. Porém, pela base de dados, o jogo atualmente possui 105 avaliações feitas pelos usuários, uma proporção de avaliações positivas de 0.3810 e, portanto, é classificado como não-positivo.

Tabela 5.21: Resultado das Aplicações dos Modelos – Geral.

Spider Queen cave	Modelagem	Base de Dados
total_reviews	245	105
user_score	0.7805	0.3810
new_score	Positiva	Não-Positivo

Action & Adventure

O jogo sorteado para a base de Ação e Aventura dos jogos *indie* foi o *Pentacore* (app_id = 2074600), lançado em 11 de Fevereiro de 2023. É um jogo pago no valor de 9.99 dólares, possui 1 idioma disponível, conta com suporte a algum tipo de controle, porém não possui *workshop* e *trading cards*. Além disso, é *singleplayer*, não possui *achievements*, *dlc* ou opção de ser jogado online.

Os modelos previram que o *Pentacore* teria 92 *reviews*, um *user_score* igual a 0.7844 e uma classificação positiva. Porém, pela base de dados, o jogo atualmente possui 6 avaliações feitas pelos usuários, uma proporção de avaliações positivas de 0.6667 e, portanto, é classificado como não-positivo.

Tabela 5.22: Resultado das Aplicações dos Modelos – Action & Adventure.

Pentacore	Modelagem	Base de Dados
total_reviews	92	6
user_score	0.7844	0.6667
new_score	Positivo	Não-Positivo

Casual

O jogo sorteado para a base Casual dos jogos *indie* foi o *Crime Scene* (app_id = 2257070), lançado em 20 de Janeiro de 2023. É um jogo pago no valor de 24.99 dólares, possui 2 diferentes idiomas disponíveis, não conta com suporte a algum tipo de controle, não possui *workshop* e nem *trading cards*. Além disso, é *singleplayer*, não possui *achievements*, *dlc* ou opção de ser jogado online.

Os modelos previram que o *Crime Scene* teria 313 *reviews*, um *user_score* igual a 0.7571 e uma classificação positiva. Porém, pela base de dados, o jogo atualmente possui 13 avaliações feitas pelos usuários, uma proporção de avaliações positivas de 0.8462 e, portanto, é classificado como positivo.

Tabela 5.23: Resultado das Aplicações dos Modelos – Casual.

Crime Scene	Modelagem	Base de Dados
total_reviews	313	13
user_score	0.7517	0.8462
new_score	Positivo	Positivo

RPG & Strategy

O jogo sorteado para a base de RPG e Estratégia dos jogos *indie* foi o *Button VR* (app_id = 2221150), lançado em 02 de Fevereiro de 2023. É um jogo pago no valor de 8.99 dólares, possui 1 idioma disponível, não conta com suporte ao controle, não possui *workshop* e nem *trading cards*. Além disso, é *singleplayer*, não possui *achievements*, *dlc* ou opção de ser jogado online.

Os modelos previram que o *Button VR* teria 72 *reviews*, um *user_score* igual a 0.6917 e uma classificação positiva. Porém, pela base de dados, o jogo atualmente possui 5 avaliações feitas pelos usuários, uma proporção de avaliações positivas de 1 e, portanto, é classificado como positivo.

Tabela 5.24: Resultado das Aplicações dos Modelos – RPG & Strategy.

Button VR	Modelagem	Base de Dados
total_reviews	72	5
user_score	0.6917	1
new_score	Positivo	Positivo

Simulation

O jogo sorteado para a base de Simulação dos jogos *indie* foi o *SpaceBourne 2* (app_id = 1646850), lançado em 17 de Fevereiro de 2023. É um jogo pago no valor de 19.99 dólares, possui 2 diferentes idiomas disponíveis, não conta com suporte a algum tipo de controle, não possui *workshop* e nem *trading cards*. Além disso, é *singleplayer*, não possui *achievements*, *dlc* ou opção de ser jogado online, porém foi lançado em acesso antecipado.

Os modelos previram que o *SpaceBourne 2* teria 1532 *reviews*, um *user_score* igual a 0.7911 e uma classificação positiva. Porém, pela base de dados, o jogo atualmente possui 772 avaliações feitas pelos usuários, uma proporção de avaliações positivas de 0.8212 e, portanto, é classificado como positivo.

Tabela 5.25: Resultado das Aplicações dos Modelos – Simulation.

SpaceBourne 2	Modelagem	Base de Dados
total_reviews	1 532	772
user_score	0.7911	0.8212
new_score	Positivo	Positivo

Sports & Others

O jogo sorteado para a base de Esportes e Outros dos jogos *indie* foi o *Hidden World 3 Top-Down 3D* (app_id = 2312500), lançado em 21 de Fevereiro de 2023. É um jogo pago no valor de 99.99 dólares, possui 103 diferentes idiomas disponíveis, não conta com suporte a algum tipo de controle, não possui *workshop* e nem *trading cards*. Além disso, é *singleplayer*, possui 6 *achievements*, não possui *dlc* ou opção de ser jogado online.

Os modelos previram que o *Hidden World 3 Top-Down 3D* teria 3945 *reviews*, um *user_score* igual a 0.6928 e uma classificação não-positiva. Porém, pela base de dados, o jogo atualmente possui 11 avaliações feitas pelos usuários, uma proporção de avaliações positivas de 0.9091 e, portanto, é classificado como positivo.

Tabela 5.26: Resultado das Aplicações dos Modelos – Sports & Others.

Hidden World 3 Top-Down 3D	Modelagem	Base de Dados
total_reviews	3 945	11
user_score	0.6928	0.9091
new_score	Não-Positivo	Positivo

6 Considerações Finais

Este trabalho teve como objetivo entender as características e os gêneros dos jogos *indie* da Steam para realizar uma análise preditiva de sua popularidade. Para isso, foram utilizadas 3 variáveis respostas: a quantidade de avaliações recebidas pelos usuários, a proporção de avaliações positivas e a descrição da pontuação das avaliações.

Os dados dos jogos foram coletados da Steam Web API, estruturados em um *dataframe* e analisados. Por terem muitos jogos *indie* sem nenhuma ou pouca avaliação, a base de interesse foi selecionada para que tivesse apenas jogos com pelo menos 5 avaliações, como também foi realizada uma segmentação temporal, tendo apenas jogos lançados entre 2018 e 2022. Na base final com 18.748 observações, um entendimento dos gêneros foi fundamental para o desenvolvimento dos modelos e, portanto, foi realizada uma análise de associação para os gêneros desses jogos, como também um agrupamento desses gêneros, utilizando agrupamento hierárquico, para serem utilizados na modelagem.

Em relação às variáveis respostas é possível observar que existe diferenciação entre os diferentes gêneros. Jogos que possuem o gênero Casual conseguem obter 73.20% dos seus jogos com classificações positivas contra 60.12% dos jogos que possuem o gênero Simulação. Além disso, algumas características dos jogos tiveram impacto em relação às variáveis respostas. Jogos que possuíam algum tipo de suporte ao controle tinham 79% dos seus jogos com avaliações positivas contra 66% dos jogos que não possuíam esse tipo de suporte. Ainda, jogos que possuíam *trading cards* ou *workshop* conseguiam obter 10 vezes mais avaliações em comparação aos jogos que não possuíam essas categorias.

Seguindo esse raciocínio, a modelagem foi realizada para entender quais variáveis eram mais importantes para cada modelo e também para cada variável resposta de popularidade. Portanto, quando comparamos os diferentes modelos, percebemos as diferentes influências das covariáveis em cada uma das variáveis respostas, e como isso muda conforme a métrica de popularidade.

Para o total de *reviews*, as variáveis com maior magnitude no modelo linear generalizado (Poisson) foram *is_free*, *singleplayer*, *multiplayer*, *is_online*, *controller_support*, *trading_cards* e *workshop*. Já no modelo Lasso (Poisson), as variáveis que não resultaram em coeficientes nulos foram *required_age*, *price_overview*, *is_online*, *trading_cards* e *workshop*. Por último, para a *Random Forest*, as variáveis *required_age*, *price_overview* e *supported_languages* foram as que tiveram maior nível de importância segundo o incremento de pureza do nó.

Para o *user score*, o número total de *achievements* foi a variável com maior nível de importância para a *Random Forest*, porém não foi significativa no modelo linear generalizado (Gamma), como também se tornou uma variável nula no modelo Lasso (Normal). E, para o *new score*, o número total de *achievements*, ter o gênero Simulação e possuir suporte ao controle foram as variáveis com maior nível de importância segundo o Índice Gini para a *Random Forest*. Porém, o *achievements* não foi significativa no modelo linear generalizado (Binomial), como também se tornou uma variável nula no modelo Lasso (Binomial).

Quando realizada a modelagem por abertura por gênero para o total de *reviews*, utilizando o modelo de *Random Forest* e o incremento de pureza do nó, algumas variáveis tiveram maior nível de importância quando comparadas entre os diferentes grupos de gêneros. Para os jogos com o gênero de Ação e Aventura, as variáveis com maior nível de importância foram *reqired_age*, *supported_languages* e *price_overview*. Para o gênero Casual, as variáveis foram *singleplayer*, *price_overview* e *workshop*. Para RPG e Estratégia, as variáveis foram *price_overview*, *supported_languages* e *required_age*. Para Simulação, a variável foi *required_age* e para Esportes e Outros foram *required_age* e *price_overview*.

Após as aplicações dos modelos nos jogos *indie* lançados em 2023, os jogos sorteados mostraram que os modelos não conseguiram detectar de forma muito precisa os resultados esperados para as métricas de popularidade. Portanto, como o objetivo desse trabalho foi entender as características e os gêneros dos jogos *indie*, extensões envolvem o uso de redes neurais e algoritmos de aprendizado profundo com foco no aumento de precisão para as variáveis respostas mencionadas. Outras possibilidades incluem coletar informações dos jogos além das que constam na Steam Web API, uma vez que ficamos limitados pelas informações fornecidas pelo serviço da plataforma.

Referências Bibliográficas

Arm (2023). Glossary - AAA Games - What are AAA Games? https://www.arm.com/glossary/aaa-games. Acessado em: Abril 2023.

Bruce, P., Bruce, A., e Gedeck, P. (2020). *Practical Statistics for Data Scientists: 50+ Essential Concepts using R and Python*. O'Reilly Media.

Chicco, D., Warrens, M. J., e Jurman, G. (2021). The Coefficient of Determination R-squared is more Informative than SMAPE, MAE, MAPE, MSE and RMSE in Regression Analysis Evaluation. *PeerJ Computer Science*, 7:e623.

Chowdhary, K. (2020). *Fundamentals of Artificial Intelligence*. Springer.

Clemen, J. (2021). Video Gaming Market Size Worldwide 2020-2025. https://www.statista.com/statistics/292056/video-game-market-value-worldwide. Acessado em: Julho 2022.

Davison, A. C. (2003). *Statistical Models*, volume 11. Cambridge University Press.

De Luisa, A., Hartman, J., Nabergoj, D., Pahor, S., Rus, M., Stevanoski, B., Demšar, J., e Štrumbelj, E. (2021). Predicting the Popularity of Games on Steam. *arXiv preprint arXiv:2110.02896*.

de Vasconcelos, L. M. R. e de Carvalho, C. L. (2018). Aplicação de Regras de Associação para Mineração de Dados na Web. *Revista Telfract*, 1(1).

Dinov, I. D. (2018). *Data Science and Predictive Analytics: Biomedical and Health Applications using R*. Springer.

Garda, M. B. e Grabarczyk, P. (2016). Is Every Indie Game Independent? Towards the Concept of Independent Game. *Game Studies*, 16(1).

Gonçalves, E. C. (2005). Regras de Associações e suas Medidas de Interesse Objetivas e Subjetivas. *INFOCOMP Journal of Computer Science*, 4(1):26–35.

Henry, C. (2011). Genre Evolution in Video Games and a Framework for Analysis. Master's thesis, University of Alberta, Faculty of Humanities Computing.

Hossin, M. e M.N, S. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5:01–11.

HTTP Statuses (2023). HTTP Statuses. https://httpstatuses.org/. Acessado em: Março 2023.

Igual, L. e Seguí, S. (2017). *Introduction to Data Science*. Springer.

James, G., Witten, D., Hastie, T., e Tibshirani, R. (2013). *An Introduction to Statistical Learning*, volume 112. Springer.

Kassambara, A. (2018). *Machine Learning Essentials: Practical Guide in R*. Sthda.

Kubat, M. (2017). *An Introduction to Machine Learning*, volume 2. Springer.

Kuhn, M., Johnson, K., et al. (2013). *Applied Predictive Modeling*, volume 26. Springer.

Li, S. Z. e Jain, A. (2009). Application Programming Interface (API). In *Encyclopedia of Biometrics*, pages 41–41. Springer US, Boston, MA.

Maleshkova, M., Pedrinaci, C., e Domingue, J. (2010). Investigating web apis on the world wide web. In *The 8th IEEE European Conference on Web Services (ECOWS 2010)*, pages 107–114. IEEE.

Metacritic (2023). How We Create The Metascore Magic. https://www.metacritic.com/about-metascores. Acessado em: Julho 2022.

Park, H. e Byun, H. (2016). Correlation Analysis: Game Professional Score and User Score on Steam. *International Journal of Multimedia and Ubiquitous Engineering*, 11(12):237–246.

Richardson, L. e Ruby, S. (2008). *RESTful Web Services*. O'Reilly Media, Inc.

Sammut, C. e Webb, G. I. (2017). *Encyclopedia of Machine Learning and Data Mining*. Springer Science & Business Media.

Steam (2023). Steam is the ultimate destination for playing, discussing, and creating games. https://store.steampowered.com/about/. Acessado em: Abril 2023.

SteamDB (2023a). Lifetime Concurrent Users on Steam. https://steamdb.info/app/753/graphs. Acessado em: Abril 2023.

SteamDB (2023b). Steam Game Releases by Year. https://steamdb.info/stats/releases. Acessado em: Julho 2022.

Steamworks (2023a). Steamworks Documentation - Steam Tags. https://partner.steamgames.com/doc/store/tags. Acessado em: Agosto 2022.

Steamworks (2023b). Steamworks Documentation - User Reviews. https://partner.steamgames.com/doc/store/reviews. Acessado em: Julho 2022.

Ting, K. M. (2017). Confusion Matrix. In Sammut, C. e Webb, G. I., editors, *Encyclopedia of Machine Learning and Data Mining*, pages 260–260. Springer US, Boston, MA.

Trnený, M. (2017). Machine Learning for Predicting Success of Video Games. Master's thesis, Masaryk University, Faculty of Informatics.

Valve (2023). At Valve we make games, Steam, and hardware. https://www.valvesoftware.com/en/about. Acessado em: Abril 2023.

VG Insights (2021). How to Estimate Steam Video Game Sales? https://vginsights.com/insights/article/how-to-estimate-steam-video-game-sales. Acessado em: Janeiro 2023.

VG Insights (2022). What can we learn from the highest earning indie developers on Steam? https://vginsights.com/insights/article/what-are-the-highest-earning-developers-doing-on-steam-that-you-arent. Acessado em: Janeiro 2023.

Wakefield, J. (2013). *Bayesian and Frequentist Regression Methods*, volume 23. Springer.

Wojtusiak, J. (2012). Machine Learning. In Seel, N. M., editor, *Encyclopedia of the Sciences of Learning*, pages 2082–2083. Springer US, Boston, MA.

Wulf, T., Possler, D., e Breuer, J. (2021). Video Game Genre (Video Games). *DOCA - Database of Variables for Content Analysis*.

Zhou, Z.-H. (2021). *Machine Learning*. Springer Nature.

Ziyang, J. (2021). Predicting the Popularity of Independent Video Games on the Steam Platform. Master's thesis, University of North Carolina, School of Information and Library Science.