Universidade Federal de Uberlândia
Faculdade de Matemática

Bacharelado em Estatística

ANÁLISE DAS CARACTERÍSTICAS DE JOGABILIDADE NO PUBG USANDO ÁRVORE DE DECISÃO

Anísio Pereira dos Santos Júnior

Uberlândia-MG
2019

Anísio Pereira dos Santos Júnior

ANÁLISE DAS CARACTERÍSTICAS DE JOGABILIDADE NO PUBG USANDO ÁRVORE DE DECISÃO

Trabalho de conclusão de curso apresentado à Coordenação do Curso de Bacharelado em Estatística como requisito parcial para obtenção do grau de Bacharel em Estatística.

Orientadora: Maria Imaculada de Sousa Silva

Uberlândia-MG

2019

Universidade Federal de Uberlândia
Faculdade de Matemática

Coordenação do Curso de Bacharelado em Estatística

A banca examinadora, conforme abaixo assinado, certifica a adequação deste trabalho de conclusão de curso para obtenção do grau de Bacharel em Estatística.

Uberlândia, ____ de _____ de 20____

BANCA EXAMINADORA

_____

Maria Imaculada de Sousa Silva

_____

Leandro Alves Pereira

_____

Ednaldo Carvalho Guimarães

Uberlândia-MG

2019

Agradecimentos

Primeiramente gostaria de agradecer a Deus que sempre me acompanhou em todas as minhas conquistas e me deu forças para conseguir chegar até aqui.

Quero agradecer ao meu pai Anisio e a minha mãe Joelia por tudo, pelo carinho, pela oportunidade que me ofereceram, pelo apoio que me fez chegar onde conseguir estar hoje.

Agradeço também ao meu padrinho Zé que me deu a motivação para sair de Sete Lagoas e ir para Uberlândia fazer o curso de estatística, mesmo sem conhecer o curso e nem a cidade, hoje percebo que foi uma das melhores decisões que já tomei.

Agradeço à professora do ensino fundamental, Alcina, que me fez gostar de matemática na sexta série e me ajudou a conseguir a medalha nas Olimpíadas de Matemática, que certamente influenciou minha escolha por esse caminho.

Agradeço à profa. Dra. Maria Imaculada por toda o apoio e ajuda como orientadora do TCC.

A todos os meus amigos que conheci na faculdade, e que ainda estão presentes na minha vida.

E finalmente agradeço à minha namorada Brenda, que sempre esteve comigo ao longo de toda essa jornada, me apoiando, me ajudando em todos os momentos. Da conquista desse diploma, sem dúvidas, uma parte é graças a ela.

Resumo

O presente trabalho desenvolvido com a base de dados disponíveis pelo Kaggle, tem como objetivo identificar, utilizando o modelo da Árvore de Decisão (AD), um compilado de estratégias utilizadas por jogadores que possuem números de vitórias significativas no modo solo do jogo online Playerunknown's Battlegrounds (PUBG). No PUBG, o modo solo é suportado por cem indivíduos em cada partida. O jogo é composto por uma ilha delimitada com áreas de morte automática pelo jogo, como também uma área circular de segurança que se estreita ao decorrer da partida, além dos demais jogadores que podem exterminar o adversário. Cada jogador poderá ter acesso a instrumentos de ataque e de proteção. O objetivo é ser o último com vida no final da partida independentemente das estratégias utilizadas. Para alcançar nosso propósito, primeiramente foi estruturado uma base literária sobre o assunto, posteriormente uma análise exploratória dos dados para verificar possíveis outliers e a distribuição das variáveis e por fim uma análise do modelo de Árvore de Decisão. Dos softwares auxiliadores, foram escolhidos Knime para análise exploratória dos dados e para a execução, validação e interpretação do modelo, os softwares Knime e R. A base foi redistribuída em 70% e 30%, para treinar o modelo e validar o mesmo, respectivamente. Usando a técnica de AD, foi possível entender o perfil dos jogadores que tendem a ter uma baixa chance de sobreviver, assim como os mais aptos, com acurácia de 0,84 para o modelo obtido.

Palavras-chave: Kaggle, Árvore de Decisão, Validação Cruzada, PUBG.

Abstract

The present work developed with the database available by Kaggle, aims to identify, using the Decision Tree (AD) model, a compiled strategies used by players who have significant wins in solo mode of the online game Playerunknown's Battlegrounds (PUBG). In PUBG, solo mode is supported by 100 players in each match. The game consists of an island delimited with areas of automatic death by the game, as well as a circular security area that narrows during the game, as well as other players that can exterminate the opponent. Each player will have access to attack and protective instruments. The goal is to be the last one alive at the end of the match regardless of the strategies used. To achieve our purpose, a literary base on the subject was first structured, then an exploratory analysis of the data to verify possible outliers and the distribution of variables and finally an analysis of the Decision Tree model. From the helper software, Knime was chosen for exploratory data analysis and for the execution, validation and interpretation of the model, the software Knime and R. The base was redistributed in 70 % and 30 %, to train the model and validate it. respectively. Using the AD technique, it was possible to understand the profile of the players who tend to have a low chance of survival, as well as the fittest, with accuracy of 0.84 for the model obtained.

Keywords: Kaggle, Decision Tree, Cross Validation, PUBG.

Sumário

Lista de Figuras

Lista de Tabelas

1. Introdução

Os esforços da indústria para criar novas tecnologias, mídias e formas de se jogar um jogo ficaram evidentes nos últimos anos, em que surgiram consoles portáteis com hardwares poderosos e controles destacáveis. No momento, pode-se acompanhar o surgimento das realidades mistas e virtuais. Entretanto, de tempos em tempos, surgem inovações no mundo dos games, que embora imprevistas, acabam sendo absorvidas rapidamente pelas indústrias criadoras de jogos. Isso já aconteceu diversas vezes dentro da história dos games, sendo que uma delas foi a revolução dos jogos MOBA (Arena de Batalha Multijogador Online). Em 2017 um novo gênero de jogo, igualmente não planejado, acabou recebendo muita atenção por parte dos players e pela indústria. Trata-se dos jogos Battle Royale [5].

Os jogos no estilo Battle Royale conquistaram o mundo. O jogo consiste de um modo “deathmatch” (conhecido em português como "mata-mata"). A partida inicia-se ao concentrar cem jogadores de mãos vazias em uma ilha, os quais devem explorar, coletar recursos e eliminar outros jogadores até que apenas um fique de pé, sendo que durante a partida, a zona de jogo encolhe ao longo do tempo [3].

Dos jogos mais populares no estilo Battle Royale, destaca-se o PUBG (PlayerUnknown’s Battlegrounds). O jogo foi lançado no final de dezembro de 2017 e testemunhou uma ascensão incomparável ao topo do mercado de videogames, tornando-se no início de 2018 o jogo mais vendido de todos os tempos [4]. O jogador começa o jogo apenas com roupas de baixo, e tem que procurar em casas abandonadas algumas vestimentas e, mais importante, armas, munição, energéticos, coletes, capacetes e kits de primeiros socorros [2]. Para se tornar campeão, o jogador deve estabelecer estratégias ao decorrer da partida para sobreviver. No PUBG, é possível escolher alguns modos de jogabilidade, entre eles pode ser escolhida a opção solo, dupla ou um esquadrão, no qual a equipe é composta por quatro jogadores.

Utilizando dados de jogadores de PUBG, o trabalho de Rokad et al [16] apresentou a aplicação de alguns modelos de aprendizado de máquina, como regressão LightGBM (Light Gradient Boosting Machine), MLP (MultiLayer Perceptron) e Floresta Aleatória. O objetivo dos autores foi comparar a eficiência dos modelos para predizer a sobrevivência no jogo em função das características de jogo adotadas, concluindo-se que todos os modelos testados podem ser usados com esta finalidade, já que apresentaram valores semelhantes de erro médio absoluto (mean absolute error -MAE). No artigo de Wei et al (2018) [18] que também realizou previsões relevantes sobre a colocação de um jogador, com o mesmo conjunto de dados, foram utilizados modelos de regressão, árvore de decisão e floresta aleatória, e modelos de misturas de normais,

apresentando como resultados alguns grupos de estratégias que levam a um bom resultado no jogo. Eles concluíram que o modelo de regressão ligth GBM apresentou melhores resultados de predição.

A base de dados para essa análise foi disponibilizada pelo Kaggle [3] que é uma plataforma de modelagem preditiva e de competições analíticas em que estatísticos e mineradores de dados competem para produzir os melhores modelos para prever e descrever os conjuntos de dados enviados por empresas e usuários.

Neste trabalho, serão utilizados dados de jogadores de PUBG, com o objetivo de aplicar o modelo de Árvore de Decisão (AD) para prever qual é a melhor estratégia de jogo que levará o jogador a ter uma probabilidade alta de vencer a partida, levando em conta as variáveis disponíveis sobre cada jogador.

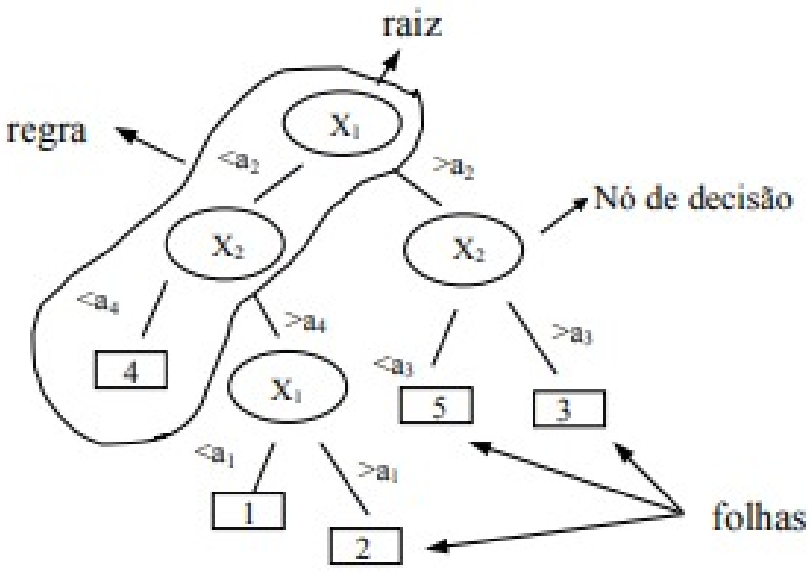
2. Fundamentação Teórica

2.1 Árvore de Decisão

De acordo com o próprio nome, cabe na demonstração visual final do tratamento de dados a partir do método não paramétrico de árvore de decisão, diversas ramificações provenientes das entradas e saídas de ambas as variáveis categóricas e contínuas introduzidas na análise. Utilizada preferencialmente como recurso em problemas de classificação e regressão, se resume em um algoritmo de aprendizagem supervisionada, geralmente com um alvo pré-definido, resultando em classes ao dividir a amostra em dois ou mais conjuntos homogêneo, redirecionados a partir dos divisores apresentados mais significativos entre as variáveis de entrada [6].

Enquanto representação de dados no meio computacional, ao instituir uma árvore nesse campo, compreende-se de forma geral, que estruturas constituídas por conjuntos de informações principais, são conhecidos como nós. Geralmente os nós representam as perguntas e suas possíveis respostas. Dentre os nós, aquele que possui característica de iniciar as ligações para outros elementos (filhos) são chamados de raiz/pai. Todo nó filho, possui nós provindos deles mesmos, aqueles nós que não apresentam filhos, são determinados como nó folha. Quando há necessidade de retirar um sub-nó de um nó filho, ocorre o processo oposto à divisão conhecido como poda. A partir do esclarecimento de toda estrutura e suas definições, pode-se resumidamente dizer que o método estatístico de árvore de decisão armazena os dados em seus nós raiz, consequentemente os nós filhos apresentam respostas para conduzir a regra, sendo que, a decisão a ser tomada será encontrada no nó folha, uma vez que, neste não se encontram nós filhos. Um exemplo de representação dessa estrutura de uma árvore de decisão pode ser visualizado na Figura 2.1. O encaminhamento do nó raiz, até o nó folha, forma o caminho para a tomada de decisão, sendo este percorrido intitulado como regra [7] [17].

Figura 2.1: Exemplo de Árvore de Decisão



Fonte: Gama, 2004 [8]

O particionamento da árvore, ocorre por meio da representação que tem atribuído em si, conduzindo a classificação. Quando houver maior representação de uma informação atribuída no nó, o atributo selecionado se distingue em atributo teste, possibilitando a geração de um novo processo de partição decorrente do atributo com maior representação.

Quando se recorre à árvore de decisão para obter resultados de classificação, os critérios para ocorrer as partições são estabelecidas por meio da entropia e índice Gini.

2.1.1 ENTROPIA

Comprende-se como entropia um recurso de medida matemática utilizado para distinguir os dados apresentados por meio da identificação da ausência de homogeneidade presente no rol de entrada combinado à sua classificação. Ou seja, em um conjunto de dados heterogêneo, a entropia é classificado como máxima (igual a 1) [17].

Dado um conjunto S, com instâncias pertencentes à classe i, com probabilidade p_i, tem-se:

$$Entropia(S) = \sum_{i=1}^{c} -p_i log_2 p_i$$

Em que:

S é o conjunto de exemplo de treino;

p_i é a proporção de dados em S que pertencem à classe i;

c é o número total de classes em S;

$Entropia(S)$ é uma medida de (não) homogeneidade do conjunto S.

A construção de uma árvore de decisão tem três objetivos, sendo eles: diminuir a entropia (a aleatoriedade da variável objetivo), ser consistente com o conjunto de dados e possuir o menor número de nós.

2.1.2 ÍNDICE GINI

Proposto em 1912 pelo estatístico italiano Conrado Gini, o índice GINI é outra medida bastante conhecida e utilizada [13]. Ele é um índice de dispersão estatística que mede a heterogeneidade dos dados e é utilizado para a seleção de atributos em que o objetivo é minimizar a impuridade [11].

O índice GINI para um conjunto de dados S, que contém n registros, cada um com uma classe C_i é dado pela equação:

$$gini(S) = i - \sum_{i=1}^{k} p_i(C_i|n)^2$$

Em que:

p_i: probabilidade relativa da classe C_i em S.

n: número de registros no conjunto S.

k: número de classes.

Se o conjunto S for particionado em dois ou mais subconjuntos S_i, O índice GINI dos dados particionados será definido pela equação:

$$gini(S, A) = i - \sum_{i=1}^{k} \frac{n_i}{n} gini(S_i)$$

Em que:

n_i : número de registros no subconjunto S_i.

n: número de registros no conjunto S.

Quando este índice é igual a zero, o conjunto de dados é puro, ou seja, todos os registros pertencem a uma mesma classe. Por outro lado, quando ele se aproxima do valor um, o conjunto apresenta os registros distribuídos igualmente entre todas as classes. Quando se utiliza o critério Gini na indução de árvores de decisão binárias, tende-se a isolar num ramo os registros que representam a classe mais frequente, assim, utilizando o atributo com menor valor do índice para a classificação, já, ao utilizar-se da entropia, balanceia-se o número de registros em cada ramo [13].

2.1.3 OVERFITTING/ SOBREAJUSTE

O processo de overfitting ocorre quando há erros no modelo ou quando o valor atribuído ao conjunto não é significativo, provocando a partição e consequentemente subconjuntos de informações desnecessárias, poluindo visivelmente o resultado final. Parte-se do princípio que os dados de treinamento precisam ser modulados quando há variáveis inéditas, afim de ofertar

à máquina um conjunto sem ruídos para que não haja erros, e assim, fluir corretamente o direcionamento da regra da árvore por meio do atributo representante [17]

Para prevenir o processo de sobreajuste é recomendado definir restrições para o tamanho da árvore, como também, se necessário, realizar o processo de podagem.

2.1.4 TAMANHO DA ÁRVORE

Para estabelecer o tamanho da árvore afim de evitar efeitos do overfitting, é imprescindível o entendimento a respeito das configurações bases da árvore. Segue abaixo cinco parâmetros delineados para definição do tamanho da árvore [6].

- **Valor Mínimo de Amostra Para Divisão de Nós**. O parâmetro, torna-se necessário para monitorar o sobreajuste e impedir que altos valores dificultem a aprendizagem do modelo sobre associações especificas, juntamente com a contribuição do recurso de validação cruzada (cross -validation).

- **Valor Mínimo de Amostra Para Nó Folha**. Nesse parâmetro, continua-se determinando uma quantidade ínfima para as amostras, porém, determinadas para o nó folha.

- **Definição do Nível de Profundidade**. O valor máximo de profundidade de uma árvore deve ser determinado como parâmetro para controlar o sobreajuste, juntamente com a validação cruzada, ao passo que permite ao modelo desenvolver a capacidade de compreender as relações mais especificas da amostra.

- **Valor Máximo de Nós Folha**. Quando se determina a quantidade total de nós folha, compreende-se que, caso seja atribuído à árvore a condição de partição binária, resultará em 2^x para profundidade "x".

- **Valor Máximo de Atributos Para Divisão**. Dentre as possíveis opções para evitar o sobreajuste, o cálculo que considera a raiz quadrada do número máximo de atributos, pode ser considerado.

2.1.5 PROCESSO DE PODAGEM

O processo de podagem pode ser recorrido em dois momentos. Quando há necessidade de pausar o modelo precocemente, chamado de pré-podagem/poda descendente, ou quando a árvore está finalizada, determinada como pós-podagem ou poda ascendente. Ambas possuem contraindicações de uso, razão esta pela qual a melhor opção será escolhida após as seguintes considerações [6]:

1. A árvore desenvolvida deverá indicar alto grau de profundidade;

2. Verificar o erro que cada nó e seus descendentes fornecerem;

3. Considerar o nó que resultar em um erro menor ou igual à soma apresentada pelos erros dos nós descendentes e transforma-lo em nó folha.

Na realização da pré-podagem, por meio da qual provoca-se o bloqueio na continuação da árvore, a partição distingue-se em confiável e não confiável. Julgando-se confiável, o crescimento é inibido, se tornando um processo mais rápido, porém, de fator menos eficiente ao considerar a possível condição de interromper erroneamente um seguimento sub-ótimo, enquanto que na pós-podagem, a árvore é desenvolvida até seu tamanho máximo, do qual a sub-árvore que indicar a operação mais sensata, será selecionada e todos os demais nós de mesmo nível hierárquico, serão podados.

2.2 Métricas de Desempenho e Matriz de Confusão

Dentro da construção de um modelo estatístico, cabe ao profissional, após o ajuste e a realização do processo adequado de poda, avaliar a capacidade discriminatória do modelo. Enquanto possibilidade de uso, tem-se a matriz de confusão, a qual proporciona via tabela, a diferenciação entre eventos e não eventos reais, com aqueles apresentados pelo modelo, baseado em quatro medidas, sendo elas [1]:

- Verdadeiro positivo (VP): considerado corretamente como evento pelo valor observado e valor predito.
- Falso positivo (FP): considerado erroneamente como evento pelo valor predito, sendo um não evento pelo valor observado.
- Verdadeiro negativo (VN): considerado corretamente como não evento pelo valor observado e valor predito.
- Falso negativo (FN): considerado erroneamente como não evento pelo valor predito, sendo evento pelo valor observado

Tabela 2.1: Matriz de confusão

	Valor Observado (valor verdadeiro)		
Valor Predito		Y = 1	Y = 0
	Y = 1	VP (verdadeiro positivo)	FP (falso positivo)
	Y = 0	FN (falso negativo)	VN (verdadeiro negativo)

Fonte: Portal Action [1]

A matriz de confusão permite calcular métricas que medem a capacidade discriminatória do modelo. Para o presente estudo, as seguintes métricas foram analisadas: Acurácia, Sensibilidade e Valor Preditivo Positivo.

A métrica Acurácia (ACC) é a proporção de predições corretas, sem considerar o que é positivo e o que negativo e sim o acerto total, ou seja, busca extrair de todas as previsões, os acertos. Sua formula é dada por:

$$ACC = (VP + VN)/(VP + VN + FP + FN)$$

A Sensibilidade (SENS) trabalha com o valor obtido pelo verdadeiro positivo, para obter a proporção dos valores apresentados tanto na predição quanto no valor real. Basicamente, presume a habilidade do modelo em determinar a amostra como evento, e dentro dos valores reais, ser um evento, ou seja, é a probabilidade de classificar como evento uma observação dado que ela realmente é um evento. Sua formula é dada por:

$$SENS = VP/(VP + FN)$$

Valor Preditivo Positivo (VPP) caracteriza-se pela relação quantitativa sobre todas as previsões positivas, em que o valor real considera a amostra como evento, e no modelo também julgou-se ser evento, ou seja, ele mede a probabilidade de uma amostra realmente ser um evento, dado que ela foi classificada como evento. A equação é dada por:

$$VPP = VP/(VP + FP)$$

2.3 Validação Cruzada

A validação cruzada é uma técnica para avaliar a capacidade de generalização de um modelo, a partir de um conjunto de dados [12]. Esta técnica é amplamente empregada em problemas em que o objetivo da modelagem é a predição. Busca-se então estimar o quão preciso é este modelo na prática, ou seja, o seu desempenho para um novo conjunto de dados.

Existem vários tipos de métodos de validação cruzada (LOOCV – validação cruzada de saída única [14], o método holdout [12], validação cruzada k-fold [12]). Para essa pesquisa foi utilizado o método de validação cruzada K-Fold.

K-Fold ilustrado na Figura 2.2, consiste basicamente nas etapas abaixo:

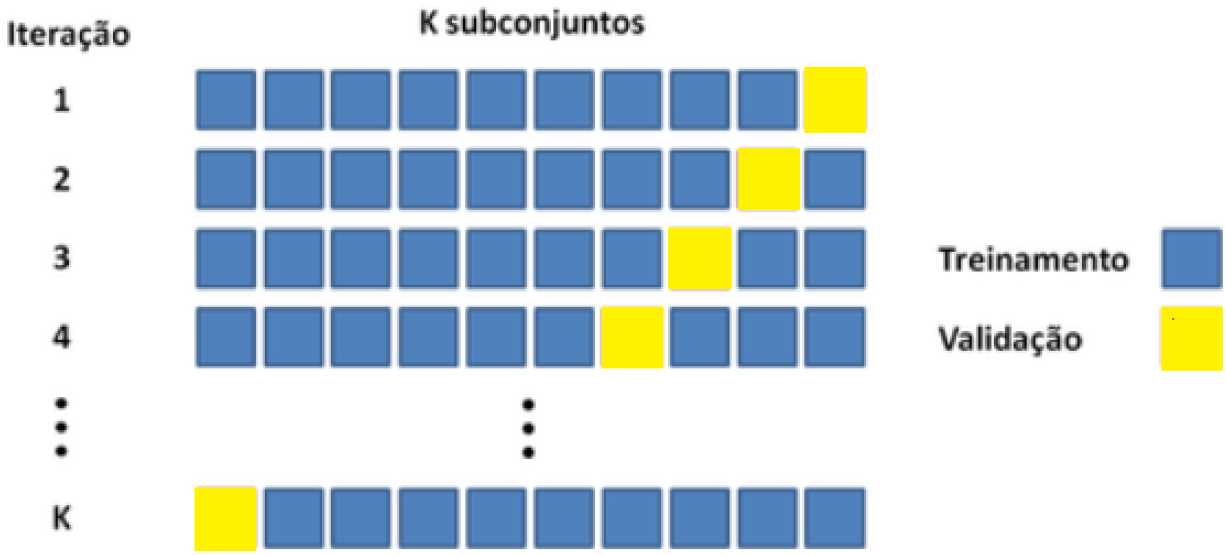
1. Divida aleatoriamente os dados em k subconjuntos, também chamados de dobras.

2. Encaixe o modelo nos dados de treinamento (ou k-1 dobras).

3. Use a parte restante (dobra não usada no treinamento do modelo) como conjunto de testes para validar o modelo. (Normalmente, nesta etapa, a precisão ou o erro de teste do modelo é medido).

4. Repita o procedimento k vezes. Ao final das k iterações calcula-se a acurácia sobre os erros encontrados, anteriormente, obtendo-se assim uma medida mais confiável sobre a capacidade preditiva do modelo.

Figura 2.2: Validação Cruzadapelo método K-Fold



Fonte: Eric Couto [1]

Para essa pesquisa o K foi igual a 10 e para cada etapa da validação foi usada a taxa de acerto (acurácia) como etapa de validação do modelo.

3. Metodologia

3.1 Base de Dados

Os dados obtidos no Kaggle contém estatísticas anônimas de jogadores do jogo PUBG e informações de quantas pessoas o jogador eliminou, quantas armas, energéticos e kits de primeiros socorros que ele usou, etc. Além disso a base contém a informação da classificação final de cada jogador em uma partida.

No jogo PUBG, existem atualmente três tipos de categorias de partidas sendo elas: Solo, Duplas ou Squads. No estilo de partida solo, todos os jogadores se enfrentam (cada um por si). No estilo duplas, cada jogador possui um parceiro dentro da partida. No estilo Squads, são formados equipes de 4 jogadores. Cada categoria de partida, requer um perfil de estratégia diferente, como por exemplo, em Duplas ou Squads, os jogadores conseguem ajudar seus companheiros de equipe quando eles estão morrendo, o que não acontece nas partidas Solo. Portanto o objetivo da pesquisa é traçar um perfil de estratégia no modo solo removendo da análise o histórico de partidas Duplas ou Squads.

A base contém 720713 registros de jogadores anônimos em partidas solo de PUBG. Para a análise dos perfis dos jogadores, 14 variáveis foram utilizadas sendo elas:

- boosts: Número de itens de reforço usados;
- damageDealt: Dano total causado.
- headshotKills: Número de jogadores inimigos mortos com tiros na cabeça;
- heals: Número de itens de cura usados;
- kills: Número de jogadores inimigos mortos;
- killStreaks: Número máximo de jogadores inimigos mortos em um curto período de tempo;
- longestKill: Maior distância entre o jogador e o oponente no momento em que este foi morto;
- rideDistance: Distância total percorrida em veículos medida em metros;
- roadKills: Número de mortes em um veículo;
- swimDistance: Distância total percorrida pela natação, medida em metros;

- vehicleDestroys: Número de veículos destruídos;

- walkDistance: Distância total percorrida a pé, medida em metros;

- weaponsAcquired: Número de armas coletadas;

- winPlacePerc: Colocação final do jogador em uma partida, em que 1 corresponde ao primeiro lugar e 0 corresponde ao último lugar da partida;

A base de dados foi separada em duas partes sendo 70% destinado para o treinamento do modelo de AD e 30% para validação dos mesmos.

Todo o processo de manipulação dos dados foi realizado no software knime [9]. Já a construção e a validação do modelo foram realizados nos softwares Knime [9] e R [15].

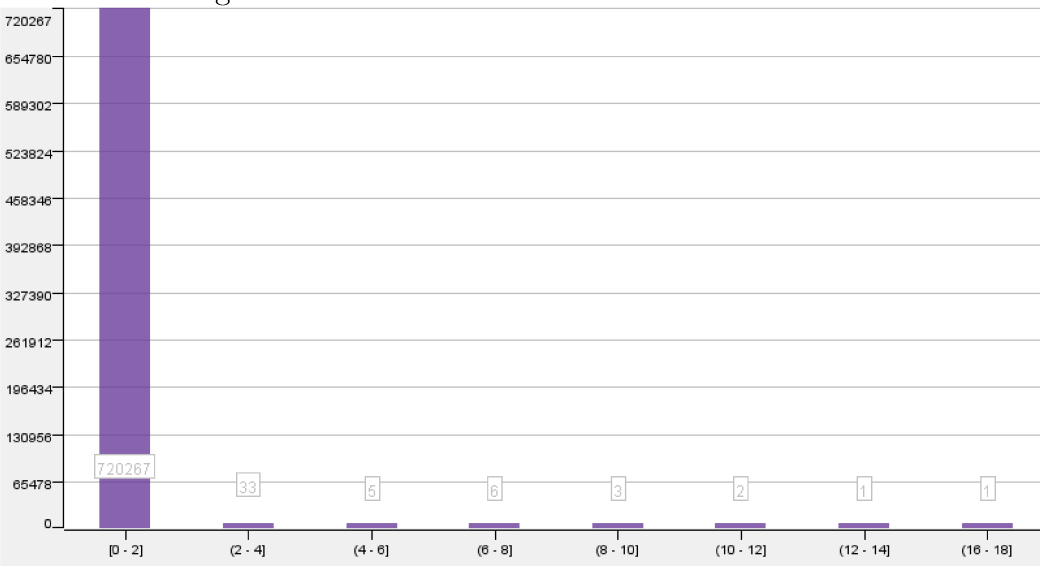
3.2 Organização dos Dados para Análise

Foram analisados algumas linhas do conjunto de dados que apresentam valores discrepantes (*outliers*). Em alguns casos dos jogos online, os jogadores podem usar algumas estratégias ou apresentar características que podem ser consideradas como anomalias ou trapaças no jogo. Considerando o objetivo de identificar melhores estratégias legais pra se vencer o jogo, os casos identificados como possíveis anomalias ou trapaças, descritos a seguir, foram eliminados do banco de dados.

- **Matar Sem Se Movimentar**: Um tipo de trapaça muito comum em PUBG é um jogador conseguir abater oponentes sem se mover durante uma partida. A base continha 394 casos dessa anomalia e todos foram excluídos.

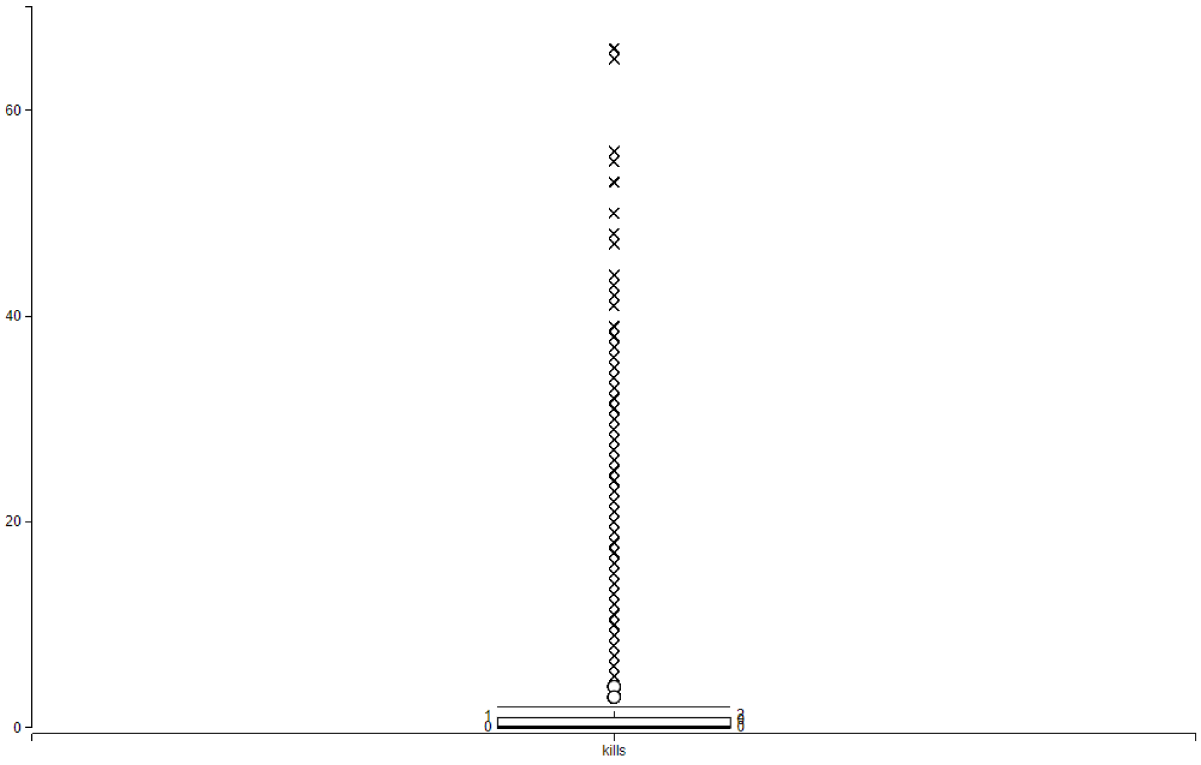
- **Matar Usando Um Veículo**: Abates usando um veículo (Figura 3.1) são muito comuns no Battle Royalle, porém muitas mortes já são um caso suspeito de trapaça. Portanto, todos os jogadores que possuem mais de 10 mortes usando veículos foram removidos totalizando 4 registros.

Figura 3.1: Número de mortes usando um veículo



- **Muitos Abates**: Para verificar esse tipo de anomalia, primeiramente foi realizado um boxplot (Figura 3.2) para analisar a variável "kills". No estilo de jogo "solo", ter um número muito grande de abates em uma partida com 100 jogadores é um caso suspeito de trapaça. Observa-se na figura 3.2 que alguns jogadores durante uma partida tiveram mais de 60 abates. Filtrando jogadores onde a porcentagem de registros foi muito pequena (acima de 30 mortes), encontramos 45 registros que também foram removidos da análise.

Figura 3.2: Total de abates



- **Mortes de longa distância**: A maioria das mortes são feitas a uma distância de 100 metros ou mais próxima. No entanto, existem algumas que podem ser consideradas

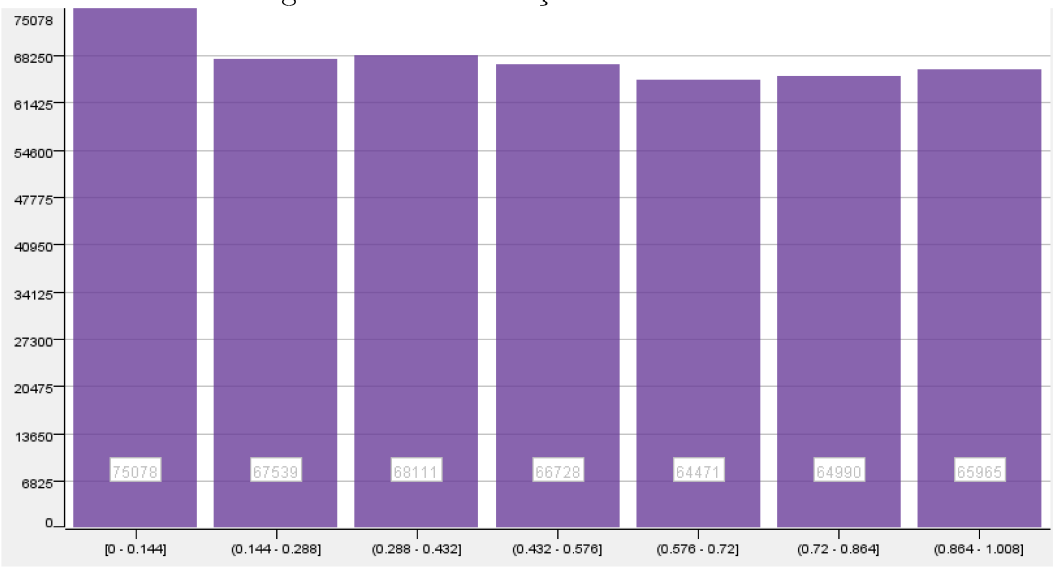
outliers, já que matam a mais de 1 km de distância. Provavelmente isso ocorre devido a métodos ilegais realizados por jogadores. Três registros que foram removidos.

No total, 446 registros fora removidos da base de treinamento.

3.3 Encontrando os grupos de classes ideais

A variável alvo (winPlacePerc) é uma variável contínua que se encontra dentro do intervalo de 0 a 1. Pode-se observar na Figura 3.3 que winPlacePerc possui uma distribuição uniforme, isso se justifica pois em cada partida, haverá uma pessoa que ficou em primeiro lugar, outra em segundo, outra em terceiro e assim sucessivamente até o último colocado. Uma alternativa para este tipo de variável seria adotar um modelo de regressão Beta, proposto por Ferrari e Cribari-Neto (2004) [10], que permite explorar o efeito de covariáveis sobre uma variável resposta que assume valores entre zero e um. Neste caso, uma limitação do modelo seriam os extremos, já que a variável alvo assume também os extremos zero e um.

Figura 3.3: Distribuição da variável Alvo



Para este trabalho foi escolhido o modelo de classificação de Árvore de Decisão sendo que algumas análises foram executadas no software Knime e para a execução do modelo final foi utilizado o software R.

Para obter um melhor resultado do modelo, a variável resposta que inicialmente apresentava valores dentro do intervalo de 0 a 1 foi transformada em classes ou categorias. Inicialmente a variável alvo foi dividida em duas classes, em que a observação foi classificada como sendo "Baixa"chance de vencer, se o valor for menor que 0,5 e classificada como "Alta"chance de vencer, se o valor for maior ou igual a 0,5. Em seguida, o modelo de AD foi gerado pelo software KNIME utilizado as configurações padrões do software. Todas as 13 variáveis explicativas foram utilizadas para o modelo. Podemos observar na Tabela 3.1 que o modelo apresentou um bom desempenho em relação ao VPP e a sensibilidade.

Tabela 3.1: Desempenho do Modelo de AD dentro do intervalo estipulado

Intervalo	VPP	Sensibilidade
0,0 |- 0,5	0,914	0,909
0,5 |- 1,0	0,906	0,911

Entretanto, essa divisão realizada não está muito adequada para interpretação, pois um perfil que classifica o indivíduo como “Alta” chance de vencer, sendo que ele se encontra entre os 50 a 100 melhores, não é adequado. Portanto foram realizadas algumas análises medindo diferentes tipos de intervalos e verificando o VPP e a sensibilidades dos modelos AD. Observando todos os resultados gerados Tabelas 3.2, 3.3, 3.4 e 3.5, o intervalo mais adequado por possuir um melhor VPP e Sensibilidade, foi definido com as seguintes especificações: "Baixa"se o valor for menor ou igual a 0,10, "Média"se for entre 0,10 e 0,75 e "Alta"caso o valo seja maior ou igual a 0,75 (Rabela 3.5).

Tabela 3.2: Desempenho do Modelo de AD dentro do intervalo estipulado

Intervalo	VPP	Sensibilidade
0,0 |- 0,90	0,942	0,974
0,90 |- 1,0	0,692	0,495

Tabela 3.3: Desempenho do Modelo de AD dentro do intervalo estipulado

Intervalo	VPP	Sensibilidade
0,0 |- 0,75	0,931	0,939
0,75 |- 1,0	0,881	0,791

Tabela 3.4: Desempenho do Modelo de AD dentro do intervalo estipulado

Intervalo	VPP	Sensibilidade
0,0 |- 0,50	0,911	0,912
0,50 |- 0,75	0,609	0,611
0,75 |- 1	0,806	0,801

Tabela 3.5: Desempenho do Modelo de AD dentro do intervalo estipulado

Intervalo	VPP	Sensibilidade
0,0 |- 0,10	0,746	0,733
0,10 |- 0,75	0,873	0,882
0,75 |- 1	0,811	0,796

4. RESULTADOS

4.1 VERIFICAÇÃO DO SOBREAJUSTE DO MODELO

A princípio 70% da base de dados foi utilizada para a etapa de construção do modelo (base de treinamento). Para a validação cruzada (VC), optou-se por utilizar o valor de K igual a 10, e assim cada um dos 10 ajustes utilizou 90% das observações para treinamento do modelo e avaliou os 10% restantes como base de teste. A medida de performance foi a taxa de acerto (acurácia). Para essa etapa da VC, todas as 13 variáveis explicativas foram utilizadas em um modelo de árvore de decisão definida com as configurações padrões do software KNIME. Os resultados da taxa de acerto (acurácia) em cada parte da VC são apresentados na Tabela 4.1. O modelo de Árvore de Decisão em todas as 10 etapas da VC teve uma taxa de acerto superior a 84%.

Observa-se na Tabela 4.1 que de acordo com a VC estima-se, no geral, quase sempre o mesmo valor de acurácia, concluindo-se que não há sobreajuste (Overfitting) pois a acurácia em todas as etapas se manteve homogênea.

Tabela 4.1: Acurácia gerada pelo modelo de Árvore de decisão

Etapas da VC	**acurácia (%)**
1	84,294
2	84,910
3	84,503
4	84,446
5	84,643
6	84,391
7	84,564
8	84,218
9	84,670
10	84,298

O estudo da Validação Cruzada, soma das 10 etapas, demonstrou resultados satisfatórios (Tabela 4.2), com valores elevados de acurácia (84,4%), de sensibilidade (79,7% para a classe "Alta" e 88,2% para a classe "Média") e de VPP (81,5% para a classe "Alta" e 87,4% para a classe média. Assim realizou-se novamente a execução do modelo de AD para encontrar e

executar o ponto de poda ideal.

Tabela 4.2: Resultados das métricas geradas pelo modelo de AD

	Classes		
Métricas	**Alta**	**Média**	**Baixa**
Acurácia		0,844	
Sensibilidade	0,797	0,882	0,74
Especificidade	0,94	0,779	0,967
VPP	0,815	0,874	0,745

4.2 Critério de Poda

Para encontrar o ponto de poda ideal, foi gerado inicialmente uma árvore completa definindo o parâmetro de Complexidade (CP) como sendo zero (nenhuma penalidade resultando em uma árvore totalmente crescida). A Figura 4.1 mostra o valor de cada erro (X-val) relacionado a cada etapa do tamanho da árvore. A princípio, à medida que cresce o tamanho da árvore, o X-val, vai reduzindo gradativamente. Após um certo ponto do tamanho da árvore (aproximadamente 155), o erro começa a subir gradativamente, o que indica que árvores com tamanhos muito grandes, para essa base, tendem a ter um valor de erro maior. Assim, a árvore foi podada de modo que o erro fosse o mínimo observado, resultando em um nível de profundidade da Árvore com valor de 154 e o CP = 4.625454e−05. A ilustração da Árvore final geral após a poda, pode ser visualizada na Figura 4.2

Figura 4.1: Gráfico de parâmetros de complexidade de poda para uma árvore totalmente crescida.

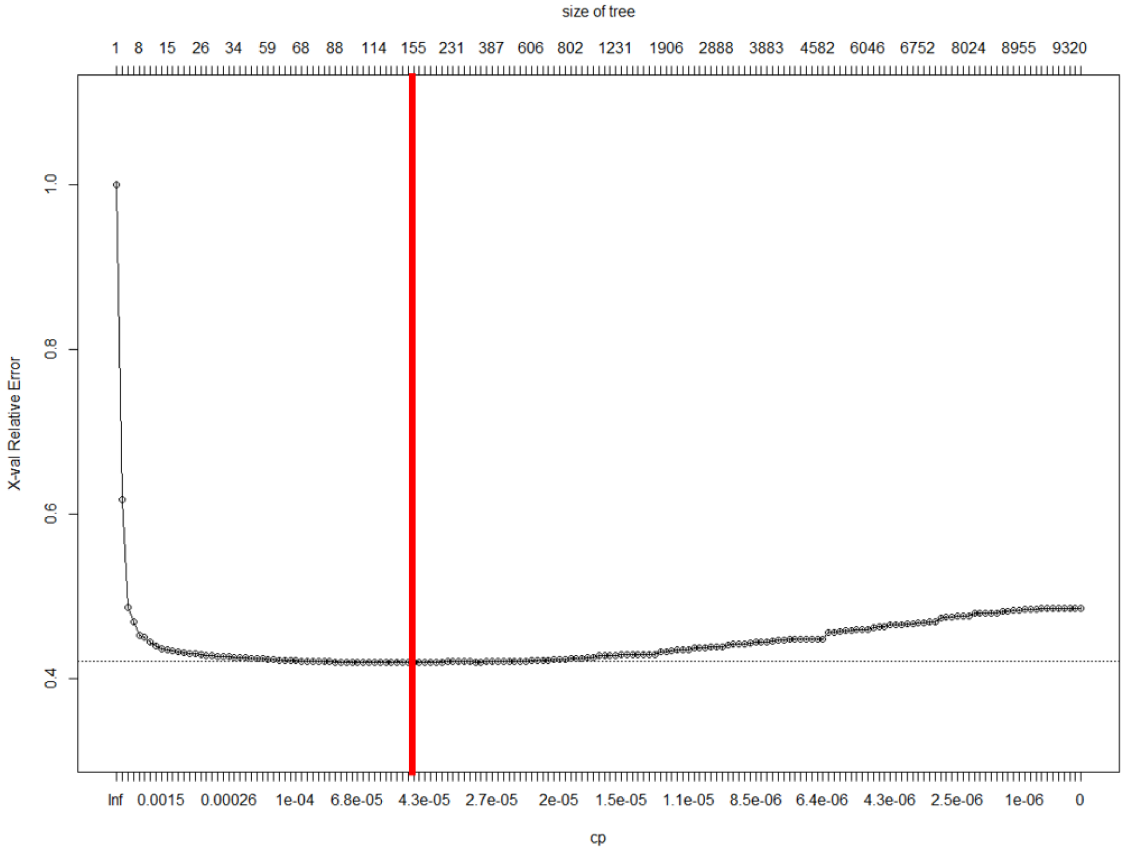
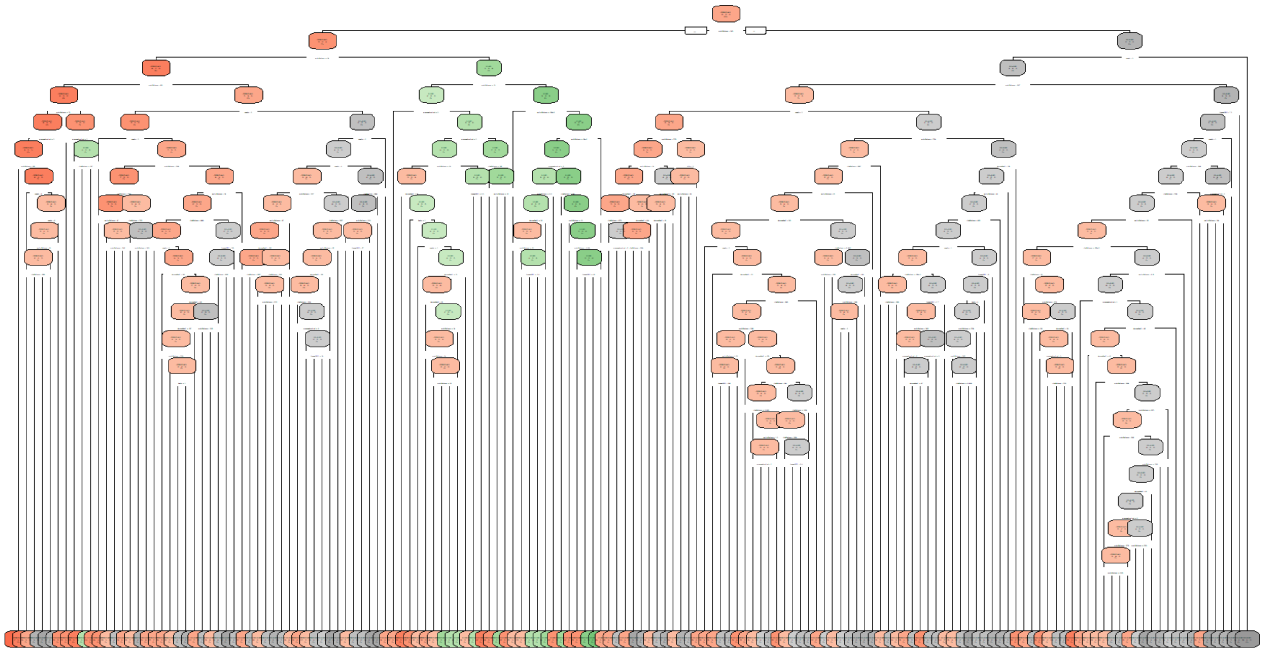


Figura 4.2: Imagem da árvore gerada após a poda.



4.3 VALIDAÇÃO DO MODELO SELECIONADO NA BASE DE TESTE

Nessa etapa foi utilizado a AD para avaliar na base de teste (30% da base separada na etapa inicial), a eficiência do modelo ajustado. Os resultados das métricas geradas se encontram na Tabela. 4.3. Observa-se que a Acurácia para a base de teste foi de 84.30% (bem próximo da acurácia da etapa da VC que foi de 84,40%). As demais métricas (Sensibilidade, Especificidade e VPP) também tiveram um desempenho bem similar à etapa da VC. Portanto, há uma forte indicação de que o modelo consegue ter uma boa previsão para novos dados (dados que não foram treinados pelo modelo).

Tabela 4.3: Resultados das métricas do modelo na base de Teste

Métricas	**Alta**	**Média**	**Baixa**
Acurácia		0,843	
Sensibilidade	0,795	0,881	0,747
Especificidade	0,939	0,782	0,966
VPP	0,81	0,872	0,758

Verificou-se na matriz de confusão (Tabela 4.4), resultante da base de teste, os valores preditos pelo modelo para cada uma das divisões da variável alvo consideradas na etapa inicial da análise. Na Tabela 4.4, os números de acertos na classificação podem ser observados na diagonal principal, enquanto os erros podem ser verificados fora da diagonal principal. Do total de 53064 registros classificados como melhores jogadores pelo modelo, temos que 43177 foram classificados como “Alta” chance de vencer, ou seja, foram classificados corretamente, e desse total, 9880 foram, classificados erroneamente como "Média"chance de vencer, e 7 foram classificados erroneamente como “Baixa” chance de vencer.

Tabela 4.4: Matriz de confusão

Real/Predito	Alta	Média	Baixa
Alta	43177	10338	268
Média	9880	119390	5990
Baixa	7	6921	20108

4.4 INTERPRETAÇÃO DO MODELO

Na Tabela 4.5, são apresentados os resultados da importância das 13 variáveis preditoras na geração do modelo de AD. A variável "walkDistance"foi a que teve maior importância (51%), o que indica que teve uma forte correlação com a variável alvo ("winPlacerPerc"). Já as variáveis "vehicleDestroys"e "roadKills"tiveram uma baixa correlação com a variável alvo indicando que essas duas variáveis não são importantes para predizer a variável "winPlacerPerc".

Tabela 4.5: Importância das variáveis para o modelo de AD

Abreviação	Atributos	Importância (%)
X1	walkDistance	50,720%
X2	boosts	15,612%
X3	longestKill	7,535%
X4	heals	7,134%
X5	damageDealt	6,443%
X6	kills	6,284%
X7	weaponsAcquired	4,538%
X8	headshotKills	0,579%
X9	swimDistance	0,456%
X10	killStreaks	0,348%
X11	rideDistance	0,343%
X12	roadKills	0,004%
X13	vehicleDestroys	0,003%

Com a execução do algoritmo AD foram geradas regras ou padrão dos dados. Para cada regra identificada foi apresentada a probabilidade de um jogador, que nas condições da regra, foi classificado na categoria indicada. Na Tabela 4.6, estão apresentadas 3 regras nas quais tem-se uma alta probabilidade de o jogador ficar na categoria "Baixa". Observa-se na regra 2 que para um jogador que percorrer a pé uma distância entre 18 e 39 metros, não encontrar nenhuma arma, percorrer nadando uma distância menor que 16 metros e percorrer, usando veículos, uma distância menor que 370 metros, a probabilidade de o jogador ser classificado na categoria "Baixa"é de 75%.

Tabela 4.6: Regras para a classe com Baixa chance de vitória

Regras	Acurácia da regra (%)	X1	X7	X9	X11
1	86	0		0	
2	75	18-39	0	< 16	< 370
3	73	$< 8{,}5$	> 1	0	

Na Tabela 4.7, encontram-se os melhores jogadores ("Alta"). Observando-se a regra 5, tem-se um perfil em que o jogador percorre a pé uma distância entre 1200 e 1400 metros, usa entre 1 a 3 itens de reforço, a morte com maior distância do jogador está acima de 160,4 metros e percorre uma distância menor do que 39 metros nadando e maior que 49000 metros usando veículos. Esse perfil é caracterizado no jogo como "Sniper"que são os jogadores que costumam conseguir abates a longa distância e tem probabilidade de 82% de ser classificado na categoria dos melhores jogadores. Esta regra sugere então que o uso de veículo, evitando confronto direto e matando a longa distância, pode conduzir a um perfil vencedor.

Tabela 4.7: Regras para a classe com Alta chance de vitória

Regras	Acurácia da regra (%)	X1	X2	X3	X5	X9	X11
4	97	> 2368	2-4				
5	82	1200-1400	1-3	$> 160{,}4$		< 39	> 49000
6	74	1400-1800	0		> 434	< 34	

No perfil com "Média"chance de vencer (Tabela 4.8), em que na regra 7 a variável X5 (dano total causado) apresenta um valor elevado (de 259 a 263) caracterizando um jogador que durante uma partida, não evita o conflito direto com outros adversários, e que usa de 2 a 3 itens de reforço, percorrendo uma distância alta usando veículo. Comparando-se o perfil dessa regra com o perfil da regra 6, em que o jogador também apresentou valor ainda mais alto de dano total causado, porém não percorreu distância de carro, mas foi classificado como alta chance de vitória, tem-se indícios de que o uso do veículo associado com perfil agressivo pode diminuir a chance de sucesso nos confrontos.

Tabela 4.8: Regras para a classe com Média chance de vitória

Regras	Acurácia da regra (%)	X1	X2	X5	X7	X9	X11
7	78	1500-1700	2-3	259-263		< 61	< 3900
8	74	75-960			0		

Na Tabela 4.7 verifica-se por exemplo na regra 4 que jogadores que percorrem a pé uma distância maior que 2368, e que usam entre 2 e 4 itens de reforço, tem uma probabilidade de 0,97 de serem classificados na categoria dos melhores jogadores. Resultado semelhante foi encontrado por Wei et al (2018) [18], que definiram algumas estratégias indicadas pra ser um vencedor no jogo. Estes autores estimaram que um jogador que tenha estratégia mais agressiva, que não evite confrontos tem uma posição média no jogo de 0,95, enquanto um jogador mais conservador tem posição média de 0,4. Eles concluíram ainda que evitar confrontos, ou ficar parado no jogo, tem classificação média de apenas 0,26, sugerindo-se então que para ser um perfil o vencedor, o jogador não deve estar parado, também concordando com a regra 4 deste trabalho, que reforça um perfil de movimento ou de percorrer longa distância.

5. Conclusões

O modelo de árvore de decisão estimado apresentou resultados coerentes e boa capacidade de discriminação dos jogadores nas categorias propostas. A técnica da validação cruzada permitiu concluir que houve um bom ajuste aos dados, e por meio da base de dados de teste, foi confirmada a alta capacidade de acerto, mesmo nos dados não treinados pelo modelo, com uma acurácia acima de 0,84.

Usando a técnica de AD, foi possível entender o perfil dos jogadores que tendem a ter uma baixa chance de sobreviver, assim como os mais aptos. Identificou-se que a variável mais correlacionada com chance de vitórias é a variável distância percorrida, inferindo-se que um jogador que tenha percorrido uma grande distância tem grandes chances de vitória. O jogador agressivo, que provoca grandes danos, não evita confronto direto percorrendo grande distância utilizando veículo, tem menos chance de ser vencedor do que um jogador agressivo, mas que evita confronto direto, prefere abater a longa distância e não utiliza veículo.

Referências Bibliográficas

[1] *Action, P.: Análise de regressão - Predição, 2019.* http://www.portalaction.com.br/analise-de-regressao/45-predicao.

[2] *O que é PUBG e porque você deveria jogar (com dicas para iniciantes).* https://www.androidpit.com.br/pubg-o-que-e-dicas-porque-jogar.

[3] *PUBG Finish Placement Prediction (Kernels Only).* https://www.kaggle.com/c/pubg-finish-placement-prediction.

[4] *PUBG se torna o jogo mais vendido da história no PC.* http://gamepress.com.br/pubg-se-torna-o-jogo-mais-vendido-da-historia-no-pc/.

[5] *A revolução dos jogos Battle Royale.* https://revolutionnow.com.br/a-revolucao-dos-jogos-battle-royale/.

[6] *Um tutorial completo sobre modelagem baseada em árvores de decisão (códigos R e Python).* https://www.vooo.pro/insights/um-tutorial-completo-sobre-a-modelagem-baseada-em-tree-arvore-do-zero-em-r-python/.

[7] *Árvores de Decisão.* https://medium.com/machine-learning-beyond-deep-learning/%C3%A1rvores-de-decis%C3%A3o-3f52f6420b69.

[8] *Árvores de Decisão.* https://docplayer.com.br/23242998-Arvores-de-decisao-sumario-joao-gama-arvores-de-decisao-motivacao-construcao-de-uma-arvore-de-decisao-podar-a-arvore.html.

[9] Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., Ohl, P., Sieb, C., Thiel, K. e Wiswedel, B.: *KNIME: The Konstanz Information Miner.* Em *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)*. Springer, 2007, ISBN 978-3-540-78239-1.

[10] Ferrari, S. e Cribari-Neto, F.: *Beta regression for modelling rates and proportions.* Journal of applied statistics, 31(7):799–815, 2004.

[11] Garcia, S. C.: *O uso de árvores de decisão na descoberta de conhecimento na área da saúde.* 2003.

[12] Kohavi, R. *et al.*: *A study of cross-validation and bootstrap for accuracy estimation and model selection*. Em *Ijcai*, vol. 14, pp. 1137–1145. Montreal, Canada, 1995.

[13] Marin, M. A.: *Indução de Árvores de Decisão para a Inferência de Redes Gênicas*. Tese de Doutoramento, Universidade Tecnológica Federal do Paraná, 2013.

[14] Ng, A. Y. *et al.*: *Preventing"overfitting"of cross-validation data*. Em *ICML*, vol. 97, pp. 245–253, 1997.

[15] R Core Team: *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. http://www.R-project.org/.

[16] Rokad, B., Karumudi, T., Acharya, O. e Jagtap, A.: *Survival of the Fittest in PlayerUnknown BattleGround*. arXiv preprint arXiv:1905.06052, 2019.

[17] SILVA, L. M.: *Uma aplicação de Árvores de Decisão, Redes Neurais e KNN para a identificação de modelos ARMA Não-Sazonais e Sazonais*. Rio de Janeiro. 145p. Tese de Doutorado-Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, 2005.

[18] Wei, W., Lu, X. e Li, Y.: *PUBG: A Guide to Free Chicken Dinner*. 2018.