

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS  
ESCOLA POLITÉCNICA  
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



**MODELO PARA PREDIÇÃO DE RESULTADOS VOLTADO PARA JOGOS  
ELETRÔNICOS**

MICAEL LEVI NUNES DE SOUZA

GOIÂNIA  
2022

MICAEL LEVI NUNES DE SOUZA

**MODELO PARA PREDIÇÃO DE RESULTADOS VOLTADO PARA JOGOS  
ELETRÔNICOS**

Trabalho de Conclusão de Curso apresentado à Escola Politécnica, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação.

Orientador(a):

Prof. Me. Olegário Corrêa da Silva Neto

Banca examinadora:

Prof. Me. Eugênio Júlio M. C. Carvalho

Prof. Dr. José Luiz de Freitas Júnior

GOIÂNIA  
2022

MICAEL LEVI NUNES DE SOUZA

**MODELO PARA PREDIÇÃO DE RESULTADOS VOLTADO PARA JOGOS  
ELETRÔNICOS**

Trabalho de Conclusão de Curso aprovado em sua forma final pela Escola Politécnica, da Pontifícia Universidade Católica de Goiás, para obtenção do título de Bacharel em Ciência da Computação, em 08/06/2022.

---

Orientador(a): Me. Olegário Corrêa da Silva Neto

---

Prof. Me. Ludmilla Reis Pinheiros dos Santos  
Coordenador(a) de Trabalho de Conclusão de  
Curso

GOIÂNIA  
2022

Dedico este trabalho aos meus pais e amigos que sempre me incentivaram.

## **AGRADECIMENTOS**

Agradeço aos meus professores e colegas por me ajudarem a desenvolver este trabalho.

## RESUMO

Apresenta um modelo capaz de prever resultados em partidas competitivas de jogos eletrônicos, com foco em Counter-Strike: Global Offensive. Utilizar de dados individuais dos jogadores, para cada possível papel, de um time para construir características gerais do time e assim realizar o cálculo da probabilidade deste vencer. Calcular a média geral de cada posição, parte da base de dados foi construída utilizando de 10 partidas dos trinta melhores times do mundo, totalizando 300 partidas. A previsão é feita dando valor a qualidade dos jogadores de ambos os times, comparando com as médias gerais e simulando todos os tipos de confrontos possíveis de uma partida cinco contra cinco. A avaliação do modelo foi realizada utilizando da divisão entre o número de confrontos que um time ganhou nas simulações e o total de simulações realizadas. Ao final da pesquisa foi obtido o resultado de 69,43% de acerto, um valor considerado satisfatório para a pesquisa. A forma com que os dados foram trabalhados, permite apresentar comparações entre qualquer jogador e os melhores jogadores mundo, esta também pode ser realizada para o time como todo.

**Palavras-chave:** Previsão; Jogos eletrônicos; Trabalho acadêmico; Counter-strike:Global Offensive; CS:GO.

## ABSTRACT

It presents a model capable of predicting results in competitive electronic games matches, focusing on Counter-Strike: Global Offensive. Use individual player data, for each possible role, of a team to build general characteristics of the team and thus calculate the probability of this winning. To calculate the general average of each position, part of the database was built using 10 matches of the thirty best teams in the world, totaling 300 matches. The forecast is made by valuing the quality of the players of both teams, comparing with the general averages and simulating all types of possible confrontations of a five-vs-five match. The evaluation of the model was performed using the division between the number of matches that a team won in the simulations and the total number of simulations performed. At the end of the research, the result of 69.43% of correct answers was obtained, a value considered satisfactory for the research. The way in which the data were worked allows to present comparisons between any player and the best players in the world, this can also be carried out for the team as a whole.

**Keywords:** Prediction; Electronic games; Academic work; Counter-strike:Global Offensive; CS:GO.

## LISTA DE FIGURAS

Figura 1 — Fluxo do Web Scraping . . . . .	17
Figura 2 — Curva Logística . . . . .	22
Figura 3 — Visão Geral da Estrutura . . . . .	23
Figura 4 — Taxa de acerto dos testes . . . . .	26

## LISTA DE TABELAS

Tabela 1 — Exemplo Normalização - Dados iniciais . . . . .	19
Tabela 2 — Exemplo Normalização - Dados normalizados . . . . .	20
Tabela 3 — Tipos de variáveis da Regressão Logística . . . . .	20
Tabela 4 — Tamanho dos mapas do CS:GO . . . . .	33

## **LISTA DE ABREVIATURAS E SIGLAS**

API	Application Programming Interface
CS:GO	Counter-Strike: Global Offensive
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol

## SUMÁRIO

1	<b>INTRODUÇÃO</b>	11
1.1	OBJETIVO	12
1.1.1	<b>Objetivo Geral</b>	12
1.1.2	<b>Objetivos Específicos</b>	12
1.2	ESCOPO	12
1.3	ESTRUTURA DO TRABALHO	13
2	<b>FUNDAMENTAÇÃO TEÓRICA</b>	14
2.1	COUNTER-STRIKE: GLOBAL OFFENSIVE	14
2.1.1	<b>Papéis</b>	15
2.2	COLETA DE DADOS	16
2.2.1	<b>HLTV.org</b>	16
2.2.2	<b>Web Scraping</b>	16
2.3	ANALISADOR DE REPLAYS	18
2.3.1	<b>Arquivos de DEM</b>	18
2.4	NORMALIZAÇÃO	18
2.5	REGRESSÃO LOGÍSTICA	20
3	<b>PROCEDIMENTOS METODOLÓGICOS</b>	23
3.1	ESTRUTURA	23
3.2	COLETA DO CONJUNTO DE DADOS	24
3.3	QUAIS DADOS ANALISAR	25
4	<b>RESULTADOS</b>	26
4.1	PRIMEIRO TESTE	26
4.2	SEGUNDO TESTE	27
4.3	TERCEIRO TESTE	27
5	<b>CONSIDERAÇÕES FINAIS</b>	28
5.1	TRABALHOS FUTUROS	28
5.1.1	<b>ANALISADOR DE REPLAYS</b>	28
5.1.2	<b>TAMANHO DA BASE DE DADOS</b>	29
5.1.3	<b>CONSIDERAÇÃO DOS MAPAS</b>	29
5.1.4	<b>MAIS CARACTERÍSTICAS</b>	29
5.1.5	<b>REDES NEURAIS</b>	29
5.1.6	<b>OUTROS JOGOS</b>	30
	<b>REFERÊNCIAS</b>	31
	<b>APÊNDICE A — TABELA COM TAMANHO DOS MAPAS</b>	33
	<b>APÊNDICE B — LISTA DE DADOS UTILIZADOS</b>	34
	<b>APÊNDICE C — LISTA DE TIMES ANALISADOS</b>	36

## 1 INTRODUÇÃO

A definição de jogo, segundo Houaiss (2003) , é a junção de atividades, que submetidas a regras, definem quem vence e quem perde, tendo natureza recreativa ou competitiva.

Os jogos eletrônicos são um tipo de jogo que, de acordo com Gularte (2010), utilizam de um dispositivo eletro-eletrônico para funcionar, em grande maioria, não exigem nenhuma qualidade física, mas necessitam de coordenação, reflexos e cognição. Estes funcionam com um conjunto de instruções gravadas em uma unidade de armazenamento, como placas e cartuchos, essas instruções são processadas por um computador e controladas pelo jogador através de dispositivos como mouse, teclado, volantes e joysticks.

Probabilidade, conforme Houaiss (2003) , é o conjunto de razões que tornam algo provável.

Para a estatística, probabilidade é o quociente entre o número de casos favoráveis e o número total de casos possíveis. Nesta área, utiliza-se da teoria das probabilidades para tentar quantificar a noção de provável em determinados eventos em que é possível definir possibilidades com probabilidades (JAYNES; CROW, 1995).

Modelos de probabilidade, de acordo com Barbeta (2006) , tem como objetivo entender melhor fenômenos aleatórios, estes são construídos a partir de hipóteses ou conjecturas sobre o problema definido e podem ser dividido em duas partes:

- dos possíveis resultados (o espaço amostral) e
- de uma certa lei que nos diz quão provável é cada resultado (as probabilidades).

Nestes tipos de modelo, os testes de hipótese e estimadores estatísticos são derivados por meio de modelos estatísticos.

Existem três propósitos para um modelo estatístico, de acordo com Konishi e Kitagawa (2007) :

- Previsões;
- Extração de informação;
- Descrição das estruturas estocásticas.

Estes três propósitos são semelhantes aos propósitos apresentados por Friendly e Meyer: previsão, estimativa e descrição.

A previsão é uma tarefa que utiliza as informações disponíveis para antecipar comportamentos e eventos futuros com maior precisão possível (HYNDMAN; ATHANASOPOULOS, 2018). Esta pode ser aplicada em diversas áreas como meteorologia, administração e logística.

Como não poderia ser diferente, esta área vem se expandindo para os esportes eletrônicos, um exemplo disto é o "IA Hackathon" promovido pela empresa BLAST em 2020, para desenvolvimento de um algoritmo capaz prever qual time irá ganhar a rodada corrente de um jogo eletrônico chamado Counter-Strike: Global Offensive (CS:GO) com base em dados fornecidos em tempo real (FUSTER, 2020).

## 1.1 OBJETIVO

### 1.1.1 Objetivo Geral

O propósito desta pesquisa é prever qual time irá vencer uma partida competitiva de CS:GO, utilizando dados dos jogadores para construir dados gerais do time.

### 1.1.2 Objetivos Específicos

- Pesquisar métodos probabilísticos que possam contribuir para a elaboração do modelo;
- Determinar quais dados serão utilizados;
- Construir um modelo capaz de prever qual time irá vencer uma partida competitiva de CS:GO a partir dos resultados obtidos nos objetivos anteriores;
- Validar a assertividade do modelo com base nos resultados obtidos e resultados esperados.

## 1.2 ESCOPO

Como há uma enorme variedade de dados que podem ser obtidos de um jogo de tiro em primeira pessoa como o CS:GO, é uma tarefa de enorme complexidade analisar todos estes dados no tempo dado para a realização desta pesquisa. Sendo assim, este terá como foco construir um protótipo que possa explorar as possíveis formas de, através dos dados individuais dos jogadores, construir dados relativos ao time e assim obter a probabilidade de um time ganhar de outro.

Inicialmente, os dados para análise serão escolhidos subjetivamente, pois é a única forma de identificar quais dados são relevantes na fase inicial, porém, com o evoluir da pesquisa, alguns dados serão adicionados ou removidos, de acordo com o impacto que o mesmo tem para com o resultado.

### 1.3 ESTRUTURA DO TRABALHO

Este documento está dividido nas seguintes etapas:

- **Fundamentação teórica (capítulo 2)**, apresenta e explica os conceitos teóricos relevantes para a realização da pesquisa.
- **Procedimentos metodológicos (capítulo 3)**, descreve detalhadamente os procedimentos realizados e seus resultados.
- **Avaliação do modelo (capítulo 4)**, descreve a assertividade do modelo no decorrer das diferentes fases da pesquisa.
- **Considerações finais (capítulo 5)** referente a parte de reflexão sobre a pesquisa realizada, expõe aplicações práticas, limitações e pontos de melhoria para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 COUNTER-STRIKE: GLOBAL OFFENSIVE

Com sua primeira versão publicada em agosto de 1999, Counter-Strike é um jogo eletrônico de tiro em primeira pessoa, multijogador *on-line*, sendo febre em seu lançamento e, até hoje, um dos jogos com mais jogadores do mundo. O último título deste franquia publicado, foi o Counter-Strike: Global Offensive (CS:GO) (VALVE, 2012).

Segundo o *web site* COUNTER-STRIKE.NET (2022), existem sete modos de jogo, são eles:

- **Competitivo:** É o modo clássico do CS:GO, onde dois times de cinco jogadores competem em uma partida melhor de 30 sob as regras padrões do jogo, onde os jogadores devem comprar coletes, armas, kit de desarme e gerenciar sua economia no jogo para aumentar suas chances de vitória. Uma partida é dividida em duas metades de 15 rodadas, onde ao final da primeira metade os times trocam seus lados indo de defesa para ataque e vice-versa. Vence a partida o primeiro time que ganhar 16 rodadas, para ganhar uma rodada, quando atacando, abates os adversários ou plantar e detonar a bomba em um dos dois locais disponíveis para tal, quando defendendo, abates os adversários ou desarmando a bomba.
- **Wingman:** Modo de jogo jogado em dois times de dois jogadores em uma partida melhor de 16, com apenas um local para plantar a bomba. Como no modo competitivo, os jogadores devem comprar coletes, armas e kits de desarme com o objetivo de plantar ou desarmar a bomba.
- **Casual:** Neste modo, de possuir os mesmos fundamentos que o modo competitivo, os jogadores não precisam se comprometer tanto quanto, podem sair ou entrar da partida livremente. Neste modo os jogadores recebem, automaticamente, coletes e kits de desarme, jogando com um sistema de economia simplificado, também é desativado o dano por fogo amigo.
- **Mata-mata:** Modo de jogo que possui um ritmo mais rápido, tem como objetivo abater adversários, a quantidade de pontos ganhos por abate depende do armamento utilizado, e no decorrer da partida será dado uma pontuação bônus por abates com uma arma específica. Este modo possui um renascimento instantâneo e após o renascimento o jogador tem um tempo limitado para comprar seus armamentos. Vence quem ao final obter mais pontos até o final de um tempo determinado.
- **Corrida armada:** É um modo em que a medida que os jogadores trocam de

arma à medida que conseguem abates, também possui renascimento instantâneo. Vence o primeiro jogador que conseguir um abate com a faca de ouro.

- **Demolição:** Os jogadores se revezam em ataque e defesa, lutando por um único de bomba em uma série de mapas projetados para jogos de ritmo acelerado. Os jogadores recebem automaticamente uma arma inicial e progredem por uma sequência de armamentos ao conseguir um abate. Quanto mais abates, maior o poder de fogo de seu armamento.
- *Flying Scoutsman:* Modo de jogo de oito contra oito, os jogadores se enfrentam em um mapa com a gravidade reduzida, não penaliza os jogadores por atirar em movimento e permite o uso de apenas duas armas - rifles de precisão SSG 08 e facas. Vence a primeira equipe que ganhar 9 rodadas.

Esta pesquisa utiliza dados apenas de partidas competitivas disputadas em campeonatos profissionais, partidas essas que podem ser disputadas em formato melhor de um, melhor de dois, melhor de cinco ou, o mais comum, melhor de três.

O ponto fundamental do desenvolvimento é a fundamentação teórica. Utilize análises, pesquisas e obras de diversos autores para embasar sua pesquisa. No desenvolvimento, você deve apresentar e discutir a literatura consultada sobre o tema, descrever a pesquisa realizada de acordo com a metodologia.

### 2.1.1 Papéis

Segundo texto disponibilizado pela Dignitas (2018) , em uma partida competitiva de CS:GO, um jogador pode assumir múltiplos papéis, sendo eles:

- **Sniper:** Jogando com os rifles de precisão disponíveis no jogo, muitas vezes, apresentam um estilo de jogo mais defensivo, marcando locais que seus adversários costumam atravessar no decorrer da rodada.
- **Rifler:** Consistente com os rifles automáticos presentes no jogo, muitas vezes, é o jogador que possui mais abates do time.
- **Suporte:** Jogador que gasta boa parte de sua economia com granadas utilitárias, limitando a visão de seus adversários ou causando dano a eles através de granadas explosivas, de luz, de fumaça e coquetéis molotov.
- **Entry-fragger:** Primeiro jogador a entrar nos locais do mapa para garantir informação para o time, muitas vezes, participando da primeira morte da rodada.
- **Lurker:** Podendo jogar longe do time ou sendo o último a entrar em ação, é o jogador que surpreende seus adversários e tenta abatê-los de lugares inesperados.

## 2.2 COLETA DE DADOS

O volume e a variedade de dados é essencial para o tipo de pesquisa presente, ter uma boa fonte de dados é necessário. Os dados utilizados são de partidas competitivas de CS:GO, principalmente de times que possuem uma ótima classificação mundial. Estes dados foram adquiridos através do web site da HLTV.org.

### 2.2.1 HLTV.org

Segundo Professeur (2022), o HLTV.org surgiu com o intuito de aumentar a quantidade de pessoas que assistem partidas de CS:GO e, atualmente, é um *web site* que possui, além os *replays* que serão analisados para a obtenção dos dados escolhidos para realizar a previsão, notícias, estatísticas, entre outras informações sobre o CS:GO.

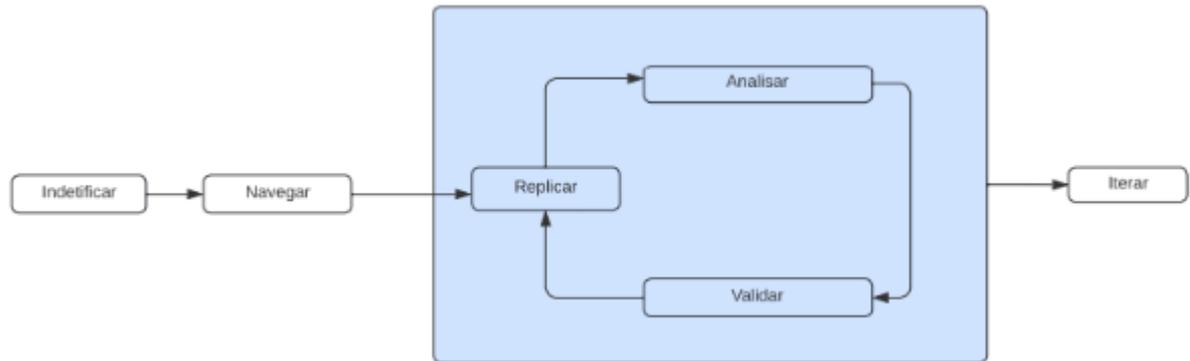
### 2.2.2 *Web Scraping*

O *web scraping*, em português coleta de dados da rede de computadores, é o ato de obter dados da rede mundial de computadores de forma automatizada, sem a utilização de uma API (*Application Programming Interface*), normalmente, navegando através do HTML (Linguagem de Marcação de HiperTexto) do *web site* para obter os dados escolhidos.

Teoricamente, *web scraping* é a prática de coletar dados por qualquer meio que não seja um programa interagindo com uma API (ou, obviamente, por um ser humano usando um navegador web). Isso é comumente feito escrevendo um programa automatizado que consulta um servidor web, requisita dados (em geral, na forma de HTML e de outros arquivos que compõem as páginas *web*) e então faz *parse* desses dados para extrair as informações necessárias. (MITCHELL, 2015, p. viii).

Segundo Lente (2018) , para entender o *web scraping*, podemos dividi-lo em um fluxo dividido em seis fases, como representado na figura 1.

Figura 1 — Fluxo do Web Scraping



Fonte: Adaptado de Lente (2018)

1. O primeiro passo do fluxo é identificar as informações que serão coletadas. Nesta parte é necessário entender bem a estrutura das páginas para elaborar um plano para extrair os dados.
2. A próxima fase é navegar, nesta parte se faz uso das ferramentas de desenvolvedor do navegador para encontrar a fonte dos dados, inspecionando, por exemplo, os elementos HTML da página ou as informações fornecidas na parte de *networking*.
3. Na fase de replicar, se tivermos que, por exemplo, fazer várias requisições HTTP para chegar até as informações escolhidas, nesta parte é importante compreender completamente os processos que a página executa para trazer o conteúdo.
4. A parte de extrair dados, também chamado de "parsear", será feita a extração de dados de um arquivo HTML, utilizando as informações obtidas na fase de navegação.
5. Na fase de validar, é necessário reproduzir o procedimento definido até agora em outras páginas semelhantes para, assim, identificar se o algoritmo está funcionando e extraindo corretamente. Em caso de falha, o processo de construção do *scraper* retorna à fase três.
6. Iterar é a última fase na produção de um algoritmo de web *scraping*, consiste em colocar o *scraper* em produção caso este tenha passado com sucesso pela parte de validação. Nesta parte, também pode ser feita a encapsulação do algoritmo em uma função que recebe uma lista de links web. Para aumentar a eficiência pode ser feita a paralelização e distribuição do *scraper*.

## 2.3 ANALISADOR DE REPLAYS

Os dados representam a maior parte da pesquisa, para obtê-los é necessário uma forma rápida de analisar e extrair estes dados dos arquivos de *replay* em formato .dem. Para essa função, foi utilizada uma ferramenta de código aberto chamada DemoInfo (2015-2022), que é capaz de navegar pelo arquivo, passando por cada *tick* do servidor, e extrair dados, como posição do jogador no mapa, arma disparada, abate, morte, assistência, entre outros. Dados estes que, posteriormente, foram utilizados para moldar métricas e alimentar o algoritmo de regressão logística para a obtenção dos resultados.

### 2.3.1 Arquivos de DEM

Arquivos DEM são arquivos de repetição do jogo utilizado por vários jogos de tiro em primeira pessoa, um dos mais populares é o CS:GO, estes são capazes de armazenar a perspectiva de um jogador que pode ser compartilhado (FILEMEMO, 2018).

Segundo File-extension.info (2022), existem dois tipos de arquivos DEM:

- Demo jogável - É um arquivo de demonstração de jogo totalmente interativa, o principal objetivo dessas demos é apresentar o jogo a seus possíveis compradores.
- Demonstração não jogável - Arquivo de demonstração em que não a possibilidade de interação, consiste em elementos de jogabilidade pré-armazenados que podem ser apresentados de várias perspectivas, este é tipicamente usado para demonstrações de desempenho do motor gráfico.

Este projeto utiliza apenas arquivos de demonstração não jogável, pois, estes arquivos são utilizados para extração de dados, não sendo necessário a interação com o ambiente do jogo.

## 2.4 NORMALIZAÇÃO

Devido à quantidade de dados selecionados para a construção do modelo, alguns possuem, naturalmente, um peso maior do que outros por, por exemplo, possuírem uma escala maior que outros. Sendo assim, foi necessária a realização da normalização dos dados.

A padronização, também chamada de normalização, põe todas as variáveis em escalas semelhantes através da subtração da média e divisão pelo desvio-padrão. Dessa forma, garantimos que uma variável não influencie exageradamente um modelo simplesmente pela escala de sua medição

original. (BRUCE; BRUCE, 2019).

Este processo de normalização é feito através da equação 1:

$$z = \frac{x - x_l}{s} \quad (1)$$

Onde:

- x: Valor a normalizar
- $x_l$ : Média aritmética da distribuição
- s: Desvio padrão da distribuição

A tabela 1 mostra os dados que serão utilizados para exemplificar como é feita normalização.

Tabela 1 — Exemplo Normalização - Dados iniciais

	Porcentagem confrontos ganhos	Total de Dano
Interz	100	100
Sh1r0	75	300

Fonte: O autor (2022)

Para realizar a normalização dos dados é necessário, primeiramente, calcular a média aritmética (equação 2) e o desvio padrão (equação 3).

$$x_l = \frac{100 + 75 + 300 + 100}{4} = 143.75 \quad (2)$$

$$s = \sqrt{\frac{(100 - 138)^2 + (100 - 138)^2 + (100 - 138)^2 + (100 - 138)^2}{4}} = 90.78 \quad (3)$$

Aplicando a fórmula de normalização nos dados da tabela 1, é obtido os resultados apresentados na tabela 2, é importante ressaltar que é considerado o valor absoluto do resultado.

Tabela 2 — Exemplo Normalização - Dados normalizados

	Porcentagem confrontos ganhos	Total de Dano
Interz	0,48	0.48
Sh1r0	0.76	1,72

Fonte: O autor (2022)

## 2.5 REGRESSÃO LOGÍSTICA

O modelo de regressão logística é um método estatístico utilizado em análises preditivas, concebido para que seja prevista a probabilidade de determinado evento ocorrer. Esta possui testes estatísticos diretos, assim compreendendo variáveis métricas e não-métricas com efeitos não-lineares. Devido ao termo de erro da variável discreta seguir a distribuição binomial, este modelo é menos afetado por dados que não foram satisfatoriamente normalizados (HAIR et al., 2008).

Este pode ser utilizado quando a variável dependente é binária, categórica ordenada ou categórica desordenada (SMOLSKI, 2019). A tabela 3 exemplifica estes três tipos de variáveis.

Tabela 3 — Tipos de variáveis da Regressão Logística

Tipo de variável	Pergunta	Possíveis Respostas
Variável dependente binária	Você votou na última eleição?	0 - Não; 1 - Sim
Variável dependente categórica ordenada	Você concorda ou discorda com o presidente?	1 - Discordo; 2 - Neutro; 3 - Concordo
Variável dependente categórica não ordenada	Se as eleições fossem hoje, em qual partido você votaria?	1 - Democratas; 2 - Qualquer um; 3 - Republicanos

Fonte: Adaptado de Torres-reyna (2014)

No exemplo da primeira coluna da tabela 3, pode-se notar que só há dois tipos de resposta possível, sendo 0 para a não presença e 1 para presença de determinada característica. Já no exemplo de variável dependente categórica ordenada é possível perceber que existe uma hierarquia determinada entre as possíveis respostas, não havendo nenhuma interseção entre elas, pois o ato de concordar e discordar são opostos e neutro não se refere a concordar ou discordar, mas, no caso apresentado, a alguém que, por exemplo, não tem uma opinião formada. O exemplo em que as respostas são do tipo categórica não ordenada percebe-se que não há algo como uma ordem ou hierarquia determinada, havendo uma interseção entre as possíveis respostas, visto que a resposta qualquer um pode ser tanto um democrata quanto um republicano.

A regressão logística a ser utilizada será com a variável de resposta dependente binária, sendo assim, os grupos de interesse são tratados com valores 0 para "não eventos" e 1 para "eventos".

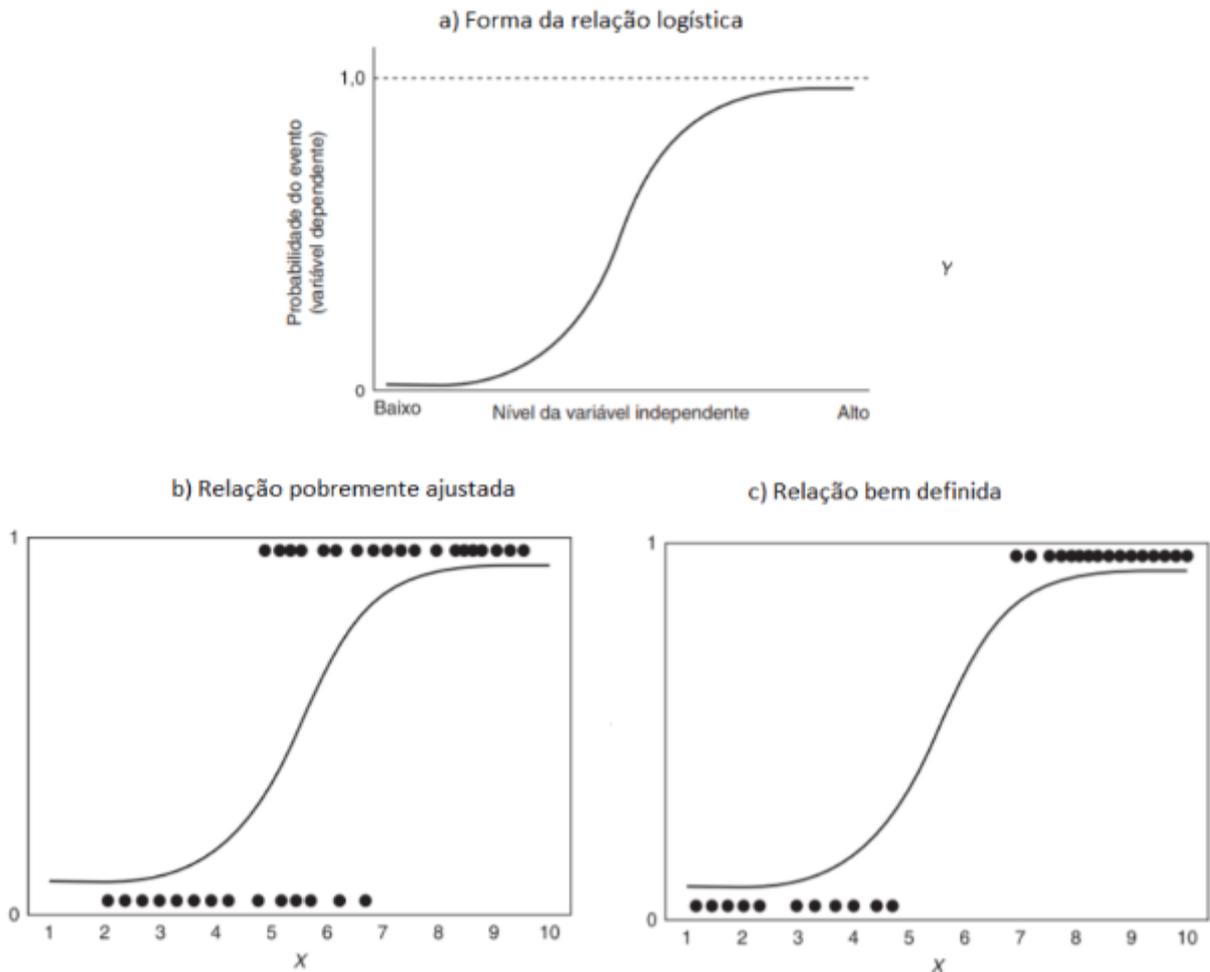
A previsão da variável dependente depende dos coeficientes logísticos e das variáveis independentes selecionadas para o modelo. Valores de probabilidade acima de 0,50 são classificados como pertencentes ao grupo de "eventos". Para estimar os coeficientes das variáveis independentes, são utilizados o valor logit ou a razão de desigualdades calculados utilizando a equação 2 (HAIR et al., 2008).

$$\text{Logit} = \ln\left(\frac{\text{prob}_e}{1 - \text{prob}_e}\right) = b_0 + b_1X_1 \dots + b_nX_n \quad (4)$$

A figura 2 representa as seguintes informações:

- a. Relação da curva logística geral.
- b. Relação de dados reais ajustada de forma pobre.
- c. Relação entre as variáveis com um bom ajuste.

Figura 2 — Curva Logística



Fonte: Adaptado de Hair et al. (2008)

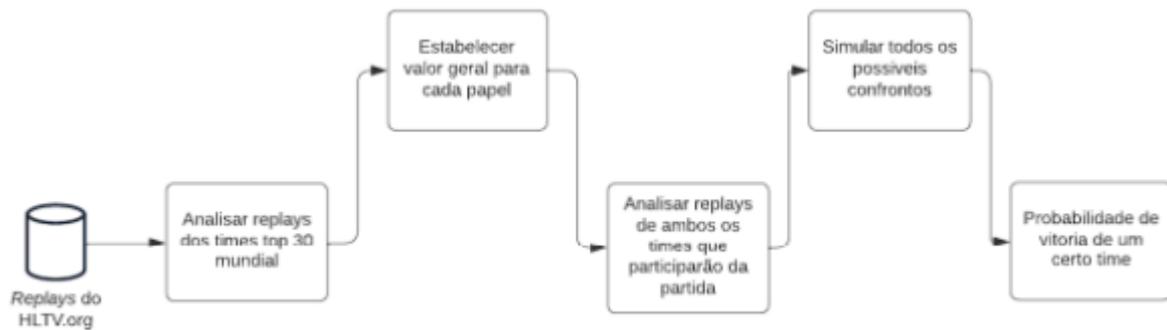
A estimação dos coeficientes é feita utilizando a máxima verossimilhança, esta que busca encontrar as estimativas com maior probabilidade dos coeficientes e maximizar a probabilidade de um evento acontecer. O valor de verossimilhança estima a qualidade que o procedimento de máxima verossimilhança se ajusta ao modelo (HAIR et al., 2008).

A qualidade do ajuste do modelo é avaliada pelo exame da previsão preditiva. O ajuste da estimação do modelo é feito através do valor  $-2$  vezes o logaritmo da verossimilhança, processo que recebe o nome de  $-2LL$ , sendo que, quanto menor este valor, melhor o modelo (HAIR et al., 2008).

### 3 PROCEDIMENTOS METODOLÓGICOS

O fluxo da figura 3 apresenta uma visão geral da estrutura do projeto e auxilia no entendimento do mesmo.

Figura 3 — Visão Geral da Estrutura



Fonte: O autor (2022)

#### 3.1 ESTRUTURA

Como visto na figura 3, o programa utiliza de *Web Scraping* (Coleta de dados Web) para baixar os *replays* dos jogos dos trinta melhores times de CS:GO, de acordo com a classificação da Electronic Sports League (ESL, 2021), disponíveis no web site da hltv.org. Estes *replays* servirão de entrada para o algoritmo de análise de *replays* descrito na seção 2.3, este analisador é encarregado de extrair as características selecionadas para realizar a previsão.

Após a obtenção dos dados necessários dos trinta melhores times, é feita a normalização destes utilizando o método descrito na seção 2.4.

Com dados normalizados, o próximo passo é calcular a média geral para cada papel, para isso é necessário:

- Calcular a qualidade dos 150 jogadores para cada papel;
- Calcular a média aritmética dos jogadores para cada papel para construir a média geral.

Com a média geral em mãos, a próxima fase é definir qual confronto será analisado, para assim, buscar as informações de cada time e, caso ainda não existam dados sobre o time e seus jogadores na base de dados, executar novamente o *scraper* e fazer a análise dos *replays* assim como na fase anterior, é importante ressaltar que, para adicionar um novo time é necessário entre 5 a 10 partidas para realizar a análise de forma consistente.

Já selecionado os dois times a serem analisados e com seus dados em mãos, da mesma forma que a parte descrita acima, é feito o cálculo da qualidade de cada jogador para todos os possíveis papéis e então, é realizada as comparações entre as médias gerais e os valores de cada jogador, esta comparação é feita para cada papel, realizando a divisão do valor da qualidade do jogador pela média geral. Esta parte é feita para, além de padronizar os dados, habilitar a possibilidade de apresentação de mais algumas informações, que são os pontos fortes e fracos do jogador em relação aos que pertencem aos melhores times do mundo. Com o valores da qualidade dos jogadores em cada papel, também é possível fazer esta mesma avaliação para o time como um todo.

Após a realização da comparação, é feita a simulação de todas as possibilidades de confrontos em um jogo cinco contra cinco, considerando que todos os dez jogadores podem exercer os cinco papéis considerados para esta pesquisa, para averiguar qual time ganha cada tipo de confronto. Com o resultado dessas simulações é calculado a probabilidade de acerto time ganhar.

Atualmente, com os resultados das simulações são utilizados para calcular a probabilidade através do resultado da divisão do número de confrontos ganhos por um dos times pelo número de tipos de confrontos analisados, porém, foi identificada de utilizar os resultados da simulação em um método probabilístico, sendo uma boa opção a utilização da regressão logística definida na seção 2.5, por permitir calcular a probabilidade de uma variável dependente, que neste caso, seria se determinado time vence uma partida ou não, através de diversas variáveis independentes, que representam os tipos confrontos, este método que também provê a possibilidade de adicionar pesos as entradas.

### 3.2 COLETA DO CONJUNTO DE DADOS

A parte da coleta do conjunto de dados foi resolvida utilizando um método de *web scraping* descrito na seção 2.2.2.

O primeiro passo é buscar o nome dos trinta melhores times do mundo dispostos no web site da ESL (2021) de forma automática, é criada uma pasta para cada time avaliado, onde serão armazenados os arquivos de formato .csv contendo todos os dados necessários para realização da análise. Após, é feita a busca do time no web site da HLTV.org e feito o download dos últimos 50 mapas jogados por estes trinta times no mês agosto de 2021 e anteriores.

A execução do *scraper* também se faz necessária em situações em que ainda não há informações sobre os times e seus jogadores na base de dados, neste caso, foram baixados os arquivos de replay das dez primeiras partidas que

ocorreram no mês de fevereiro de 2022 e anteriores.

Estes arquivos de *replay*, são utilizados como entrada para o analisador descrito na seção 2.4, para assim obter os dados necessários.

### 3.3 QUAIS DADOS ANALISAR

A seleção de quais dados analisar e avaliar se estes são realmente relevantes para a pesquisa é uma parte crucial do projeto. Para selecionar quais características considerar, inicialmente, foi feita uma análise manual de partidas de CS:GO e no decorrer do estudo, dados foram retirados, adicionados ou alterado os seus valores no intuito de validar o real impacto dos mesmos no resultado final.

Para evitar que os dados analisados fossem unicamente ligados à performance individual do jogador, foram adicionados dados relativos ao time aos papéis, como, por exemplo, o número de vezes que o time ganhou a rodada após o jogador conseguir o abate inicial. Esta medida se faz necessária para evitar que a performance individual de um jogador venha a ter um peso muito grande no resultado final. Todos os dados analisado estão descritos no apêndice B.

Entre os dados escolhidos, existem alguns com uma complexidade maior para obtenção, como por exemplo, número total de mortes e abates distantes do time, visto que os mapas possuem diferentes tamanhos, estes foram classificados em pequeno, médio e grande de acordo com o seu tamanho em unidades Hammer que, de acordo com Valve Developer Community (2018), são a medida de distância utilizada nos jogos desenvolvidos com os motores gráficos Source e Goldensource, essa distância é equivalente a uma polegada ou a três quartos de polegada. No caso de mapas, arquiteturas e alguns modelos 1 unidade Hammer é equivalente a três quartos de polegada ou a 1.905 cm (VALVE DEVELOPER COMMUNITY, 2022). Mapas pequenos foi considerada a distância de 3 kHu(unidades kilo-Hammer), mapas médios 5 kHu e mapas grandes 7.5 kHu. Os tamanhos dos mapas bem como suas classificações estão no apêndice A.

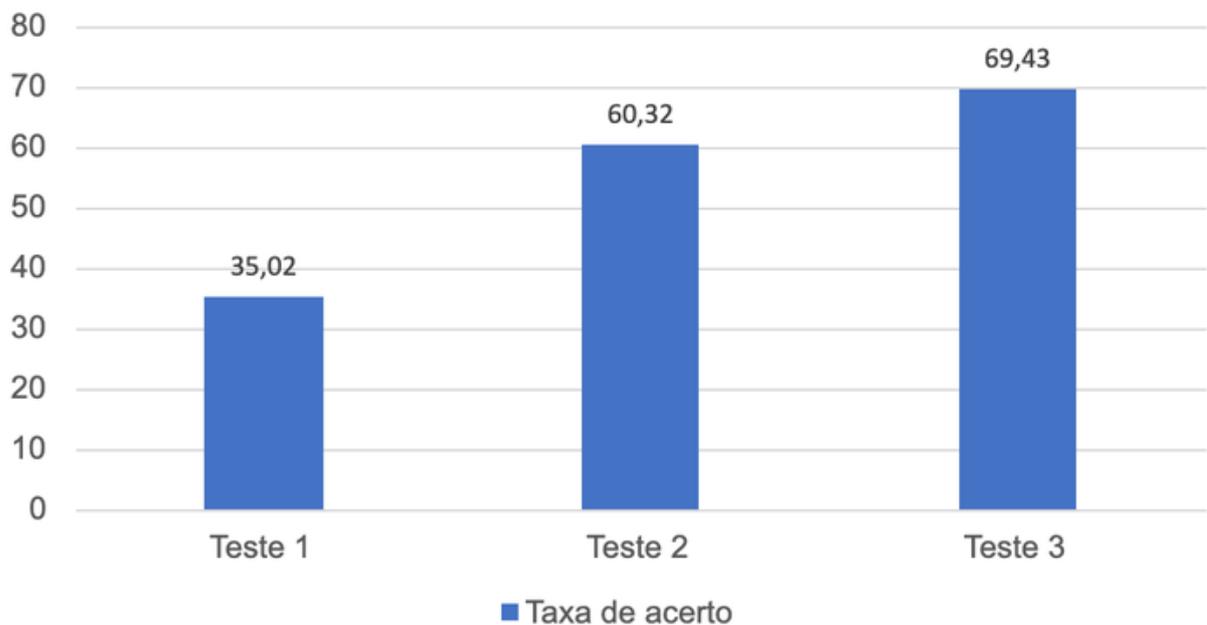
Muitos dos dados analisados foram utilizados para formar informações mais complexas, utilizando da estatística descritiva, por exemplo, a taxa de tiros acertados por um *sniper* é calculado pela divisão entre o número de tiros acertados e o número total de tiros efetuados.

## 4 RESULTADOS

Para realizar a avaliação do modelo foram utilizadas partidas de competições profissionais, do mais alto nível, que ocorreram entre janeiro de 2021 e dezembro de 2021, totalizando 494 partidas. Todos os times que foram analisados estão descritos no apêndice C.

A avaliação deste foi realizada através da comparação entre os resultados esperados e os resultados obtidos. O resultado de cada teste está representado na figura 4.

Figura 4 — Taxa de acerto dos testes



Fonte: O autor (2022)

### 4.1 PRIMEIRO TESTE

O primeiro teste foi feito nos estágios iniciais do projeto, conseqüentemente, não foram utilizadas tantas características para a realização da previsão, visto que, este tinha como objetivo se a estrutura geral era funcional. Neste teste o algoritmo conseguiu prever corretamente o resultado de 173 partidas das 494 utilizadas para avaliação, conseqüentemente, tendo assertividade de 35,02%.

Dito isto, a maior mudança feita para o próximo teste foi a adição de mais características para análise.

## 4.2 SEGUNDO TESTE

Antes de testar o algoritmo novamente, mais características foram adicionadas passando de 17 para 32 características, esta adição melhorou bastante a taxa de previsão. Consequentemente, é possível chegar a conclusão de que quanto mais tipos de dados analisados melhor.

Neste teste, o algoritmo obteve 298 acertos dos 494, com a porcentagem de acerto de 60,32%.

Nesta parte, também, foi identificada a necessidade da normalização dos dados, dado que, alguns dados possuem escaladas maiores que outros e, por resultado, têm, naturalmente, um peso maior na equação. E, outros dados não faziam sentido neste momento serem utilizados, dito que seriam interessantes para o futuro, nesta fase, não fazem diferença nesta parte do trabalho, um exemplo destes dados é a posição da mira do jogador no eixo X e Y.

## 4.3 TERCEIRO TESTE

Este foi realizado com os dados já normalizados e com mais tipos de dados adicionados, passando a ter mais de 40 características analisadas, também foram retirados os dados que não tiveram tanto impacto no modelo. Mais uma vez foi notada a melhora na taxa de acerto do algoritmo.

No ultimo teste realizado, foi obtido resultados positivos em 343 partidas das 494, tendo assertividade de 69,43%.

No decorrer deste teste, foi vista a possibilidade da implementação de uma rede neural para tornar o modelo capaz de reconhecer padrões e correlações nos dados, agrupando e classificando-os, para assim, aprender e melhorar continuamente (HAYKIN, 2000).

## 5 CONSIDERAÇÕES FINAIS

O objetivo geral deste trabalho foi prever qual time irá vencer uma partida competitiva de CS:GO, utilizando dados dos jogadores para construir dados gerais dos times, este foi atingido com sucesso, visto que, foi obtido uma taxa de acerto de 69,43%.

Para alcançar o objetivo principal desta pesquisa, foi necessário concluir os objetivos específicos, desde a pesquisa de métodos probabilísticos, onde foram selecionados a forma de normalização, a aplicação da regressão logística e a modelagem dos dados através do cálculo de médias, até a validação do modelo que foi feita utilizando da comparação entre os resultados esperados e os resultados obtidos.

Pela forma como o trabalho foi idealizado, este pode fornecer outros tipos de dados ao usuário, além da predição de vencer ou não uma partida, como a nível de certo jogador ou time em relação aos melhores jogadores e times do mundo, apresentado a eles formas de melhorar o seu nível de habilidade no jogo.

### 5.1 TRABALHOS FUTUROS

Apesar de o modelo construído apresentar resultados satisfatórios, há espaço para a realização de melhorias. Esta seção apresenta áreas onde grandes melhorias poderiam ser feitas.

#### 5.1.1 ANALISADOR DE REPLAYS

O analisador de *replays* utilizando para extrair as informações dos arquivos de demonstração, por ser de código aberto, foi de grande importância para o projeto, visto que, iria demandar muito tempo para implementar um durante o decorrer desta pesquisa. Porém este programa possui alguns problemas e bugs por ser antigo e possuir poucas melhorias implementadas ultimamente, com isso, alguns dados não puderam ser obtidos como a linha de visão dos jogadores ou informações relacionadas a granadas, especialmente com relação a granada incendiária.

Em face disto, para colocar todos os dados relevantes, se faz necessário implementar um analisador de *replays* novo ou melhorar o que foi utilizado para a construção deste projeto.

### 5.1.2 TAMANHO DA BASE DE DADOS

Mesmo que o tamanho da base de dados utilizada fosse grande, poderia ser maior. Em geral, uma base de dados maior significa uma melhor acurácia. Porém, para aumentar a base de dados há algumas barreiras, entre elas, a limitação da velocidade e da quantidade de downloads simultâneos permitidos pela HLTV.org, o tempo necessário para extração dos dados de um arquivo de demonstração e a limitação do poder computacional utilizado.

### 5.1.3 CONSIDERAÇÃO DOS MAPAS

O desempenho de um time pode variar de acordo com o mapa que está sendo jogado, visto que, é uma tarefa muito difícil para um time obter um altíssimo nível de jogo em todos os possíveis sete possíveis mapas que podem ser jogados, muitas vezes, estes, têm um mapa para banimento permanente. Seria interessante, inclusive, a construção de um algoritmo para prever quais mapas serão jogados durante uma partida para, assim, aumentar a acurácia do modelo proposto neste projeto.

### 5.1.4 MAIS CARACTERÍSTICAS

Para a realização deste trabalho, cerca de 40 características dos jogadores foram utilizadas. Com essa quantidade de dados é difícil conseguir dar um valor quantitativo para papéis e estilos de jogos mais complexos, dessa forma, se torna uma tarefa difícil, em partidas com jogadores muito habilidosos, definir as qualidades de um time e definir a probabilidade de um time vencer.

Também é interessante considerar os tipos de rodadas em que os jogadores do time possuem seus melhores números, visto que, a forma como uma rodada será jogada pelo time depende muito do estilo de jogo dos jogadores e da economia disponível para a rodada, já que a economia em jogo um dos fatores mais importantes em uma partida competitiva de CS:GO.

### 5.1.5 REDES NEURAIIS

A utilização de um algoritmo de rede neural pode ser muito aproveitada neste projeto, identificando quais confrontos são mais impactantes, qual lado, defesa ou ataque, tem maior peso para decisão de partidas, quais tipos de rodada tem mais impacto no resultado da partida, entre outros pontos que auxiliam na implementação de um sistema de pesos consistente e na melhora do modelo.

### 5.1.6 OUTROS JOGOS

Neste projeto o modelo proposto foi utilizado no CS:GO pela facilidade com que os dados podem ser extraídos dos arquivos de demonstração, todavia, os princípios utilizados neste projeto podem ser utilizados para muitos outros jogos.

Os dados poderiam ser extraídos através de algoritmos de reconhecimento de imagens e da coleta de informações, seja utilizando uma API externa, um algoritmo de *web scraping* ou de qualquer outra forma, e o programa poderia avaliar qual é o melhor time da mesma maneira.

## REFERÊNCIAS

BARBETTA, Pedro A.. **Estatística Aplicada às Ciências Sociais**. Florianópolis: Ufsc, 2006.

BRUCE, Andrew; BRUCE, Peter. **Estatística Prática para Cientistas de Dados**. Alta Books, v. 1, f. 196, 2019. 392 p.

COUNTER-STRIKE.NET. **About CS:GO**. 2022. Disponível em: <https://blog.counter-strike.net/index.php/about/>. Acesso em: 11 mai. 2022.

DIGNITAS. **Looking at Player Roles in CS:GO**. 2018. Disponível em: <https://dignitas.gg/articles/blogs/CSGO/5750/Looking-at-Player-Roles-in-CSGO>. Acesso em: 11 set. 2021.

ESL. **CS:GO WORLD RANKING**. 2021. Disponível em: <https://pro.eslgaming.com/worldranking/csgo/rankings/>. Acesso em: 23 ago. 2021.

FILE-EXTENSION.INFO. **Extensão do arquivo DEM**. 2022. Disponível em: <https://www.file-extension.info/pt/format/dem>. Acesso em: 31 mai. 2022.

FILEMEMO. **.dem Extensão de arquivo**. 2018. Disponível em: . Acesso em: 31 mai. 2022.

FUSTER, Charles. **BLAST AND SKYBOX SHOWCASES AI PREDICTION TOOL FOR CS:GO**. ESPORTZ NETWORK. 2020. Disponível em: <https://www.esportznetwork.com/blast-and-skybox-showcases-ai-prediction-tool-for-csgo/>. Acesso em: 17 ago. 2021.

GULARTE, Daniel. **Jogos eletrônicos, 50 anos de interação e diversão**. Rio de Janeiro: Novas Ideias, 2010.

HAIR, Joseph F. *et al.* **Análise multivariada de dados - 6ed**. Bookman Editora, f. 344, 2008. 688 p.

HAYKIN, Simon. **Redes Neurais - 2ed.**, f. 450. 2000. 900 p.

HOUAISS, Antonio. **Dicionário Houaiss da língua portuguesa**, f. 1461. 2003. 2922 p.

HYNDMAN, Rob J; ATHANASOPOULOS, George. **Forecasting: principles and practice**. OTexts, v. 1, f. 190, 2018. 380 p.

JAYNES, Edwin T; CROW, Wayman. **Probability Theory: The Logic of Science**. St. Louis: Washington University, 1995.

KONISHI, Sadanori; KITAGAWA, Genshiro. **Information Criteria and Statistical Modeling**. Springer Science & Business Media, f. 137, 2007. 273 p.

LENTE, Caio. **O Fluxo do Web Scraping**. curso-r. 2018. Disponível em: <https://blog.curso-r.com/posts/2018-02-18-fluxo-scraping/>. Acesso em: 17 mai. 2022.

MITCHELL, Ryan. **Web Scraping with Python**: Collecting Data from the Modern Web. "O'Reilly Media, Inc.", v. 3, f. 128, 2015. 256 p.

PROFESSEUR. **20 YEARS OF HLTV.ORG**. HLTV.org. 2022. Disponível em: <https://www.hltv.org/news/33554/20-years-of-hltvorg>. Acesso em: 27 abr. 2022.

SMOLSKI, Felipe Micail da Silva. **Software R: curso avançado**: capítulo 7 regressão logística. 2019. Disponível em: <https://smolski.github.io/livroavancado/reglog.html>. Acesso em: 21 nov. 2021.

TORRES-REYNA, Oscar. **Logit, Probit and Multinomial Logit models in R**. Princeton: Princeton University, 2014. Disponível em: <https://dss.princeton.edu/training/LogitR101.pdf>. Acesso em: 12 abr. 2022.

VALVE DEVELOPER COMMUNITY. **Dimensions**. 2022. Disponível em: <https://developer.valvesoftware.com/wiki/Dimensions>. Acesso em: 5 mar. 2022.

VALVE DEVELOPER COMMUNITY. **Hammer units**. 2018. Disponível em: [https://developer.valvesoftware.com/wiki/Hammer\\_units](https://developer.valvesoftware.com/wiki/Hammer_units). Acesso em: 21 mar. 2022.

VALVE. **Counter-Strike: Global Offensive**. Steam. 2012. Disponível em: [https://store.steampowered.com/app/730/CounterStrike\\_Global\\_Offensive/?l=portuguese](https://store.steampowered.com/app/730/CounterStrike_Global_Offensive/?l=portuguese). Acesso em: 15 abr. 2022.

**APÊNDICE A — TABELA COM TAMANHO DOS MAPAS**

Tabela 4 — Tamanho dos mapas do CS:GO

Nome do mapa	Tamanho em kHu <sup>2</sup>	Classificação
Ancient	47,19744	Pequeno
Dust2	57.84192	Médio
Inferno	54.69696	Médio
Mirage	56.93184	Médio
Nuke	80.97984	Grande
Overpass	73.9008	Grande
Vertigo	49,37281	Pequeno

Fonte: O autor (2022)

## APÊNDICE B — LISTA DE DADOS UTILIZADOS

### Para *Lurker*:

- Número mortes distante do time;
- Número abates distante do time;
- Número abates sendo o último do time vivo;
- Número mortes sendo o último do time vivo;
- Tipo do armamento usado para abater o inimigo;
- Número de vezes que o jogador exerceu papel de *lurker* na partida;
- Quantidade total de dano causado exercendo o papel de *lurker*.

### Para *Sniper*:

- Número de tiros acertados com rifles de precisão;
- Total de tiros efetuados com rifles de precisão;
- Número de vezes que o jogador exerceu papel de *sniper* na partida;
- Quantidade total de dano causado utilizando rifles de precisão;
- Número de abates conseguidos com um rifle de precisão;
- Número de abates conseguidos no local de plantio da bomba com rifles de precisão;
- Número de abates conseguidos após o plantio da bomba com rifles de precisão;
- Número de abates conseguidos com arma secundária ou faca, estando em posse de um rifle de precisão.

### Para Suporte:

- Quantidade de segundos que diminuiu o campo de visão inimigo utilizando granadas de luz;
- Quantidade de segundos que diminuiu o campo de visão inimigo utilizando granadas de fumaça;
- Quantidade de segundos que diminuiu o campo de visão aliado utilizando granadas de luz;
- Total de dano causado utilizando granada explosiva;
- Total de dano causado utilizando granada incendiária;
- Total de equipamentos não utilizados durante a rodada;
- Quantidade total de granadas usadas;
- Número total de assistências;
- Valor total de economia gasta em granadas;
- Número de vezes que o jogador exerceu papel de suporte na partida;
- Número total de participações em trocas de abates.

### Para *Rifler*:

- Número de tiros acertados com armas primarias que não rifles de precisão;
- Total de tiros efetuados com armas primarias que não rifles de precisão;
- Número de tiros na cabeça utilizando com armas primarias que não rifles de precisão;
- Quantidade total de abates utilizando com armas primarias que não rifles de precisão;
- Número de vezes que o jogador exerceu papel de *rifler* na partida;
- Quantidade total de dano causado utilizando com armas primarias que não rifles de precisão;
- Número de abates conseguidos no local de plantio da bomba com armas primarias que não rifles de precisão;
- Número de abates conseguidos após o plantio da bomba com armas primarias que não rifles de precisão;
- Número de abates conseguidos com arma secundária ou faca, estando em posse de uma arma primaria que não rifles de precisão.

Para *Entry-fragger*:

- Número de primeira morte da rodada sofrida pelo jogador;
- Número de abates no primeiro confronto da rodada;
- Número de rodadas ganhas pelo time após o jogador realizar primeiro abate;
- Número de rodadas ganhas pelo time após o jogador ter sofrido primeiro abate;
- Número de vezes que o jogador exerceu papel de *entry-fragger* na partida;
- Quantidade total de dano causado como *entry-fragger*.

**APÊNDICE C — LISTA DE TIMES ANALISADOS**

- Bad News Bears
- BIG
- Cloud9
- Complexity
- Copenhagen Flames
- Dignitas
- ENCE
- Endpoint
- Entropiq
- Eternal Fire
- Evil Geniuses
- Extra Salt
- FaZe Clan
- Fnatic
- forZe
- FunPlus Phoenix
- FURIA Esports
- G2 Esports
- Gambit Esports
- GODSENT
- Heroic
- Imperial Esports
- K23
- MIBR
- mousesports
- Movistar Riders
- Natus Vincere
- Ninja in Pyjamas
- OG
- paiN Gaming
- Renegades
- Sharks Esports
- SINNERS Esports
- SKADE
- Sprout

- Team Fiend
- Team LDLC
- Team Liquid
- Team oNe eSports
- Team Spirit
- Team Vitality
- TYLOO
- ViCi Gaming
- Virtus.pro
- Wisła Kraków



**PUC  
GOIÁS**

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS  
GABINETE DO REITOR

Av. Universitária, 1009 • Setor Universitário  
Caixa Postal 86 • CEP 74605-610  
Goiânia • Goiás • Brasil  
Fone: (62) 3946 1000  
www.pucgoias.edu.br • reitoria@pucgoias.edu.br

## RESOLUÇÃO n° 038/2020 – CEPE

### ANEXO I

#### APÊNDICE ao TCC

#### Termo de autorização de publicação de produção acadêmica

O(A) estudante Micael José Nunes de Souza  
do Curso de Ciência da Computação, matrícula 20172002800155,  
telefone: 62 998628075 e-mail michel@nunes@ig.com na qualidade de titular dos  
direitos autorais, em consonância com a Lei n° 9.610/98 (Lei dos Direitos do autor),  
autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o  
Trabalho de Conclusão de Curso intitulado  
Modelo para predição de resultados voltado para jogos  
eletrônicos, gratuitamente, sem ressarcimento dos direitos autorais, por 5  
(cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial  
de computadores, no formato especificado (Texto (PDF); Imagem (GIF ou JPEG); Som  
(WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da  
área; para fins de leitura e/ou impressão pela internet, a título de divulgação da  
produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 08 de junho de 2022.

Assinatura do(s) autor(es): Micael José Nunes de Souza

Nome completo do autor: Micael José Nunes de Souza

Assinatura do professor-orientador: Olegário Correa da Silva Neto

Nome completo do professor-orientador: Olegário Correa da Silva Neto