UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

Gabriel Herdy Rocha

PREDIÇÃO DE RESULTADOS DE PARTIDAS PROFISSIONAIS DE COUNTER-STRIKE 2 COM APRENDIZADO DE MÁQUINA

São Carlos – SP
2026

Gabriel Herdy Rocha

PREDIÇÃO DE RESULTADOS DE PARTIDAS PROFISSIONAIS DE COUNTER-STRIKE 2 COM APRENDIZADO DE MÁQUINA

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Engenharia de Computação, do Centro de Ciências Exatas e de Tecnologia, da Universidade Federal de São Carlos, como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. André Ricardo Backes.

São Carlos – SP
2026

Ficha catalográfica

Espaço reservado para inserção da ficha catalográfica emitida pela biblioteca ou pelo sistema institucional correspondente da Universidade Federal de São Carlos.

Esta página deverá ser substituída pela ficha oficial antes do depósito final do trabalho.

Gabriel Herdy Rocha

PREDIÇÃO DE RESULTADOS DE PARTIDAS PROFISSIONAIS DE COUNTER-STRIKE 2 COM APRENDIZADO DE MÁQUINA

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Engenharia de Computação, do Centro de Ciências Exatas e de Tecnologia, da Universidade Federal de São Carlos, como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação.

Aprovado em: ____/____/____.

BANCA EXAMINADORA

Orientador
Prof. Dr. André Ricardo Backes
Universidade Federal de São Carlos

Membro da banca
Edilson Reis Rodrigues Kato
Universidade Federal de São Carlos

Membro da banca
Artino Quintino da Silva Filho
Universidade Federal de São Carlos

RESUMO

Este trabalho investiga a predição do vencedor de partidas profissionais de Counter-Strike 2 usando apenas informações disponíveis antes do início do confronto. Cada partida é representada por diferenças entre os dois times, calculadas a partir de estatísticas recentes, indicadores de desempenho e uma medida de força pré-jogo. O problema foi tratado como uma tarefa de classificação binária, na qual se estima a probabilidade de vitória da equipe analisada. A metodologia separou treinamento, seleção e avaliação por ordem temporal: as partidas mais antigas foram usadas para treinamento e seleção interna, enquanto a janela 2026_s1, chamada de *holdout*, ficou reservada como avaliação posterior ao treinamento. A avaliação experimental partiu de 38 atributos pré-jogo e registrou 71 conjuntos de variáveis, 58 configurações possíveis de classificadores e 2.582 combinações efetivamente avaliadas. A escolha final foi uma regressão logística com regularização L2, $C = 0{,}3$, sem balanceamento artificial de classes, usando as 11 variáveis mais bem posicionadas pelo critério univariado. Nesse *holdout*, a regressão logística atingiu ROC-AUC de 0,6828 e acurácia de 0,6293. A avaliação prospectiva em eventos posteriores foi mantida como evidência complementar em uso operacional, sem alterar a seleção do modelo. Como contribuição, o trabalho apresenta um fluxo reprodutível para predição pré-jogo em Counter-Strike 2, conectando dados, variáveis, treinamento, comparação entre classificadores, interpretação e validação complementar.

Palavras-chave: Counter-Strike 2; esports; aprendizado de máquina; regressão logística; predição esportiva; ROC-AUC.

ABSTRACT

This work investigates win prediction in professional Counter-Strike 2 matches using only information available before each match begins. Each observation is represented by differences between the two teams, computed from recent statistics, performance indicators and a pre-match strength measure. The task is framed as binary classification, with the aim of estimating the win probability of the analyzed team. The methodology separates training, selection and evaluation in temporal order: older matches are used for training and internal selection, while the 2026_s1 window is reserved as a later-period holdout evaluation. The experimental evaluation started from 38 pre-match attributes and registered 71 feature sets, 58 possible classifier configurations and 2,582 combinations actually evaluated. The selected solution is an L2-regularized logistic regression with $C = 0.3$, no artificial class balancing and 11 variables chosen by a univariate selection criterion. On the holdout set, this logistic regression achieved ROC-AUC of 0.6828 and an accuracy of 0.6293. The prospective evaluation over later events was kept as complementary evidence from operational use, without changing the selected model. The contribution is a reproducible workflow for pre-match prediction in Counter-Strike 2, connecting data, variables, training, classifier comparison, interpretation and complementary validation.

Keywords: Counter-Strike 2; esports; machine learning; logistic regression; sports prediction; ROC-AUC.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 Introdução

Os esportes eletrônicos, também chamados de *esports*, formam um setor de entretenimento digital organizado em torno de torneios, equipes profissionais, transmissões ao vivo, patrocínios e comunidades internacionais de espectadores. Um relatório de mercado de 2022 estimava a receita global do cenário profissional em aproximadamente US$ 1,38 bilhão naquele ano e projetava crescimento para cerca de US$ 1,86 bilhão em 2025; o mesmo relatório estimava a audiência global de *esports* em 532 milhões de espectadores em 2022 (Newzoo, 2022). Além da audiência, competições profissionais envolvem premiações, contratos, calendário competitivo e incentivos de desempenho, aspectos que também aparecem em estudos sobre torneios *online* e presenciais em Counter-Strike (Shenkman; Molodchik; Parshakov, 2025). Esse contexto ajuda a explicar por que partidas competitivas passaram a ser tratadas como um domínio relevante para análise de dados e modelos preditivos.

Dentre os vários jogos utilizados em esportes eletrônicos, o Counter-Strike 2 se destaca por ser um jogo competitivo por equipes em que duas formações de cinco jogadores disputam rodadas sucessivas, alternando objetivos de ataque e defesa. As equipes precisam decidir como ocupar espaços do mapa, usar granadas, preservar ou comprar equipamentos e administrar a economia interna de cada rodada. Nesse cenário, desempenho individual, estratégia coletiva, contexto do torneio, preparação tática e variações de forma recente se combinam. Estudos sobre predição em Counter-Strike mostram que esse tipo de problema pode ser tratado com aprendizado de máquina a partir de dados históricos e estatísticas agregadas (Švec, 2022; Broms; Nordansjö, 2024). Ainda assim, antes de uma partida, a informação disponível é incompleta: mapa, veto, pressão competitiva e momento individual nem sempre estão representados nos dados.

A partir desse cenário, este trabalho propõe e avalia um fluxo de aprendizado de máquina para estimar a probabilidade de vitória em partidas profissionais de Counter-Strike 2. A abordagem é pré-jogo: a previsão não usa informações que só seriam conhecidas durante ou depois da partida. Cada observação representa um confronto e cada variável corresponde a uma diferença entre os times, como diferença de taxa recente de vitórias, diferença de indicadores médios dos jogadores e diferença de ELO.

O ELO, nome associado ao sistema de pontuação proposto por Arpad Elo originalmente para o xadrez, é uma medida numérica de força relativa (Elo, 1978). De forma resumida, uma equipe ganha ou perde pontos de acordo com seus resultados e com a força esperada do adversário; vencer um adversário forte tende a aumentar mais a pontuação do que vencer um adversário fraco. Na base construída, o valor usado em cada partida é sempre calculado antes do confronto, evitando vazamento

de informação futura.

A metodologia separa treinamento, seleção e avaliação por ordem temporal: as partidas mais antigas são usadas para treinar e comparar candidatos, enquanto a janela 2026_s1, chamada de *holdout*, fica reservada como avaliação posterior ao treinamento. Essa separação reduz o risco de avaliar o classificador em partidas usadas no ajuste dos parâmetros e torna a avaliação mais próxima de um uso futuro. A pergunta que orienta o trabalho é: *qual configuração de variáveis e classificador oferece uma predição pré-jogo competitiva, interpretável e reprodutível para partidas profissionais de Counter-Strike 2, respeitando a ordem temporal dos dados?*

Com essa pergunta, o recorte adotado deixa de ser apenas verificar se existe informação útil nos dados pré-jogo. O foco passa a ser organizar um protocolo completo de classificação binária supervisionada, com atenção à construção da base, à seleção de atributos, à comparação entre classificadores e à reprodutibilidade dos resultados.

1.1 Contribuições do trabalho

A contribuição em Engenharia de Computação está em organizar e documentar o caminho dos dados até o modelo final de previsão. O trabalho reúne coleta de estatísticas públicas de jogadores e partidas, preparação da base, construção dos atributos, seleção de variáveis, comparação de classificadores e interpretação do modelo em um fluxo que pode ser conferido e reproduzido.

De forma resumida, as principais contribuições são: a organização de uma base de dados de partidas profissionais representadas por diferenças entre equipes; a definição de atributos pré-jogo ligados a força histórica, forma recente e desempenho agregado dos jogadores; a comparação sistemática de classificadores sob o mesmo protocolo de avaliação; e a escolha de uma solução final interpretável, baseada em regressão logística, acompanhada de análises de robustez e avaliação prospectiva complementar.

1.2 Objetivos

O objetivo geral deste trabalho é treinar, avaliar e documentar um classificador binário capaz de estimar a probabilidade de vitória em partidas profissionais de Counter-Strike 2 usando apenas variáveis disponíveis antes do confronto.

Os objetivos específicos são:

- construir uma base de dados em que cada partida seja representada por diferenças entre as equipes;

- organizar atributos pré-jogo relacionados a força competitiva, forma recente e desempenho agregado dos jogadores;
- avaliar diferentes subconjuntos de variáveis, classificadores e hiperparâmetros sob o mesmo protocolo experimental;
- selecionar uma configuração final que combine desempenho, interpretação e reprodutibilidade;
- registrar limitações, robustez e avaliação prospectiva.

2 Fundamentação Teórica e Trabalhos Relacionados

2.1 Esportes eletrônicos e Counter-Strike 2

Esportes eletrônicos são competições organizadas em jogos digitais. Em cenários profissionais, equipes treinam, disputam campeonatos, recebem premiações e deixam registros digitais sobre partidas, jogadores e resultados. Trabalhos de predição em jogos competitivos usam justamente esse tipo de registro como base para tarefas supervisionadas, como discutido por Hodge et al. (2021). Essa disponibilidade de dados não elimina a incerteza do jogo, mas permite construir representações pré-jogo auditáveis.

No Counter-Strike 2, duas equipes de cinco jogadores disputam mapas formados por rodadas curtas. Em cada rodada, uma equipe atua no ataque e tenta cumprir o objetivo do mapa, enquanto a outra defende áreas estratégicas ou busca eliminar os adversários. Entre as rodadas, há uma economia interna: os times decidem quando comprar armas e utilitários, quando economizar recursos e como adaptar a estratégia ao placar. Por isso, o desempenho observado em uma partida resulta da combinação entre mira, posicionamento, uso de recursos, tomada de decisão e execução coletiva.

Trabalhos de análise em Counter-Strike mostram que ações, estados de jogo e estatísticas agregadas podem ser organizados como dados para estimar desempenho ou probabilidade de vitória (Xenopoulos; Doraiswamy; Silva, 2020; Xia; Salinas; Morstatter, 2025). No recorte deste trabalho, essa ideia aparece em indicadores como *rating*, KAST e *impact*. O *rating* é um indicador agregado de desempenho individual. O KAST resume a porcentagem de rodadas em que um jogador conseguiu abater, assistir, sobreviver ou ser trocado por um companheiro. O *impact* busca medir ações de maior influência, como abates de abertura e situações decisivas. Esses indicadores não explicam toda a complexidade do jogo, mas oferecem sinais úteis para uma representação pré-jogo.

As partidas profissionais também podem ser disputadas de forma *online* ou em LAN, quando as equipes jogam presencialmente em um ambiente local de torneio. Essa diferença de contexto ajuda a interpretar análises posteriores por tipo de partida, pois torneios presenciais tendem a envolver pressão competitiva, estrutura, premiação e nível de adversários diferentes dos confrontos disputados remotamente (Shenkman; Molodchik; Parshakov, 2025).

Também há mudanças de *metagame*, isto é, mudanças nas estratégias mais comuns ou mais eficientes em determinado período competitivo. Por isso, uma previsão construída com dados históricos precisa ser avaliada com cuidado: o modelo pode aprender padrões úteis, mas não deve ser tratado como determinístico. Essa incerteza

justifica o uso de métricas probabilísticas e de separação temporal na avaliação.

2.2 Trabalhos relacionados

Hodge et al. (2021) estudam predição ao vivo em jogos competitivos, usando dados detalhados do andamento da partida. O estudo é relevante porque trata a previsão em jogos digitais como um problema supervisionado e probabilístico. A diferença para este trabalho está no momento da previsão: aqui, o objetivo é estimar a probabilidade de vitória antes do jogo começar, sem variáveis observadas durante a partida.

Björklund et al. (2018) investigam a predição de resultados em CS:GO usando dados extraídos de demos da FACEIT e redes neurais. Esse tipo de dado é mais detalhado que o usado aqui, pois pode capturar informações internas das partidas. A abordagem adotada segue outro caminho: utiliza agregados pré-jogo e busca uma solução tabular mais simples de auditar.

Švec (2022) também aborda a predição de resultados em Counter-Strike com aprendizado de máquina e utiliza indicadores ligados à força relativa das equipes, como o ELO. Neste trabalho, essa ideia é adaptada para uma representação por diferenças entre equipes, combinando ELO pré-jogo, taxa recente de vitórias e estatísticas agregadas dos jogadores em cada recorte temporal.

Entre os trabalhos levantados, o estudo de Broms e Nordansjö (2024) é o mais próximo do recorte adotado aqui, pois discute predição de partidas de Counter-Strike com modelos como regressão logística e *Random Forest*, além de comparações com *odds*. Em relação a esse estudo, este trabalho se diferencia por conduzir uma avaliação própria, com separação temporal, seleção de variáveis e escolha documentada de uma solução final interpretável.

Outra linha de trabalhos explora Counter-Strike a partir de informações internas da partida. Xenopoulos, Doraiswamy e Silva (2020) analisam ações de jogadores em CS:GO a partir de mudanças na chance de vitória durante a partida, enquanto Xia, Salinas e Morstatter (2025) tratam o jogo como domínio para análise e modelagem preditiva estruturada. Esses estudos reforçam que o jogo possui sinais quantificáveis, embora trabalhem com níveis de detalhe e momentos de previsão diferentes do recorte pré-jogo adotado aqui.

A Tabela 1 sintetiza esses trabalhos e explicita como cada um se conecta ao recorte adotado.

Tabela 1 – Síntese dos trabalhos relacionados

Trabalho	Contexto	Relação com este trabalho
Björklund et al. (2018)	CS:GO, demos e redes neurais.	Usa dados mais detalhados; este trabalho prioriza agregados pré-jogo e rastreabilidade tabular.
Švec (2022)	Counter-Strike e aprendizado de máquina.	Dialoga com o uso de indicadores de força relativa; este trabalho usa diferenças entre equipes e agregados pré-jogo.
Broms e Nordansjö (2024)	Counter-Strike, regressão logística, *Random Forest* e *odds*.	É próximo ao problema, mas aqui o foco está em avaliação própria, separação temporal e solução interpretável.
Xenopoulos et al. (2020)	CS:GO, ações de jogadores e probabilidade de vitória.	Mostra a força de dados estruturados no jogo, mas usa informação interna da partida.
Xia et al. (2025)	CS:GO, análise e modelagem preditiva.	Reforça Counter-Strike como domínio de predição; este trabalho foca o cenário pré-jogo.
Hodge et al. (2021)	Predição ao vivo em jogos competitivos.	Situa a tarefa como aprendizado supervisionado, embora em outro jogo e com outro momento de previsão.

Fonte: elaboração própria.
Nota: A tabela resume apenas os trabalhos usados para contextualizar o problema e diferenciar o recorte adotado.

Em conjunto, os trabalhos relacionados mostram que a predição em jogos competitivos já foi estudada sob diferentes estratégias. O recorte deste trabalho é mais específico: usar informações pré-jogo de Counter-Strike 2, organizar os dados como diferenças entre equipes e comparar classificadores sob um protocolo temporal reproduzível.

2.3 Aprendizado supervisionado e classificação binária

Aprendizado de máquina supervisionado é uma abordagem em que um algoritmo aprende a partir de exemplos que já possuem entrada e saída conhecidas. Em uma tarefa supervisionada, o conjunto de treinamento apresenta ao modelo tanto as variáveis de entrada quanto o resultado correto esperado, permitindo que ele estime

uma regra de decisão para novos casos (Mitchell, 1997). Neste trabalho, cada exemplo é uma partida já encerrada: a entrada reúne as variáveis pré-jogo do confronto, como a diferença de ELO entre as equipes, e a saída conhecida é o resultado observado, vitória ou derrota da equipe analisada. Aprender, nesse contexto, significa ajustar os parâmetros do classificador para que as probabilidades previstas se aproximem dos desfechos dessas partidas passadas. O valor do ajuste, porém, está na generalização: depois do treinamento, o modelo recebe uma partida que não participou do ajuste, representada pelas mesmas variáveis, e estima a probabilidade de vitória. É essa necessidade de funcionar em confrontos não vistos que motiva a separação temporal entre treino e avaliação adotada neste trabalho.

Como existem apenas duas classes possíveis para a equipe analisada, vitória ou derrota, o problema é tratado como classificação binária. A saída probabilística é importante porque uma partida pode ser equilibrada. Em vez de apenas dizer qual time deve vencer, o modelo estima uma probabilidade, permitindo diferenciar previsões fortes de previsões próximas de 50%.

Foram avaliados classificadores supervisionados implementados em scikit-learn (Pedregosa et al., 2011). A regressão logística modela diretamente a probabilidade da classe positiva (Hosmer; Lemeshow; Sturdivant, 2013). O *K-Nearest Neighbors* (KNN), formalizado por Cover e Hart (1967), classifica um exemplo observando os exemplos mais próximos no conjunto de treino. O Gaussian Naive Bayes usa uma hipótese probabilística simples sobre a distribuição das variáveis, abordagem discutida por Hand e Yu (2001). O *Support Vector Machine* (SVM), proposto por Cortes e Vapnik (1995), busca uma fronteira de separação entre classes. A *Random Forest*, proposta por Breiman (2001), combina várias árvores de decisão para capturar relações não lineares.

A forma clássica da regressão logística estima a probabilidade da classe positiva por meio da função logística (Hosmer; Lemeshow; Sturdivant, 2013):

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)}}. \tag{1}$$

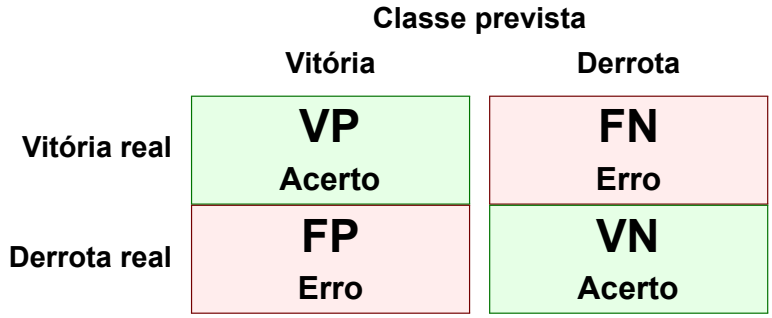
Na Equação (1), $x_1, \ldots, x_p$ representam as variáveis explicativas, β_0 é o intercepto e $\beta_1, \ldots, \beta_p$ são os coeficientes estimados. A regularização L2, usada no modelo final, penaliza coeficientes muito grandes e ajuda a estabilizar o ajuste quando há variáveis correlacionadas.

2.4 Métricas de avaliação

A avaliação de um classificador probabilístico não deve depender de uma única métrica. Em uma partida equilibrada, duas previsões podem gerar a mesma classe final e ainda assim ter níveis de confiança muito diferentes. Por isso, foram acompanhadas métricas de decisão, de ordenação e de qualidade probabilística.

Métricas de decisão. As métricas de decisão partem da matriz de confusão, que cruza a classe observada com a classe prevista pelo modelo. Na Figura 1, a classe de interesse é a vitória da equipe analisada. Assim, VP indica uma vitória prevista corretamente e VN indica uma derrota prevista corretamente. FP ocorre quando uma derrota é classificada como vitória, enquanto FN ocorre quando uma vitória é classificada como derrota. A diagonal principal reúne os acertos, VP e VN, e as demais células reúnem os dois tipos de erro, FP e FN.

Figura 1 – Estrutura conceitual da matriz de confusão para a classe vitória.



Fonte: elaboração própria.

Com um limiar de decisão fixado, a acurácia mede a proporção total de classificações corretas:

$$\text{Acurácia} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{VN} + \text{FP} + \text{FN}}. \tag{2}$$

A acurácia não é capaz de descrever sozinha o tipo de erro cometido. Por isso, precisão, revocação e F1 foram usadas como complemento:

$$\text{Precisão} = \frac{\text{VP}}{\text{VP} + \text{FP}}, \qquad \text{Revocação} = \frac{\text{VP}}{\text{VP} + \text{FN}}, \tag{3}$$

$$F1 = 2 \cdot \frac{\text{Precisão} \cdot \text{Revocação}}{\text{Precisão} + \text{Revocação}}. \tag{4}$$

No contexto desta tarefa, a precisão indica, entre as partidas previstas como vitória da equipe analisada, quantas realmente terminaram em vitória. A revocação indica quantas vitórias reais foram recuperadas pelo classificador. O F1 resume essas duas leituras em uma única medida, sendo útil quando se quer observar o equilíbrio

entre falsos positivos e falsos negativos.

Ordenação probabilística. ROC-AUC (*Receiver Operating Characteristic – Area Under the Curve*) avalia a capacidade de ordenação do modelo entre exemplos positivos e negativos. Em termos práticos, ela verifica se partidas que terminaram em vitória da equipe analisada tendem a receber probabilidades maiores do que partidas que terminaram em derrota. A curva ROC relaciona a taxa de verdadeiros positivos e a taxa de falsos positivos conforme o limiar de decisão varia. Assim, o ROC-AUC não depende apenas do limiar 0,5 e ajuda a comparar classificadores quando o interesse não está restrito a uma única regra de corte. Fawcett (2006) apresenta a análise ROC como uma ferramenta apropriada para esse tipo de comparação.

Qualidade das probabilidades. Também foram usadas log-loss e Brier score, pois ambas avaliam a qualidade das probabilidades, não apenas a classe final. Considerando p_i como a probabilidade prevista para a classe positiva e $y_i \in \{0, 1\}$ como o resultado observado, a log-loss é definida por:

$$\text{Log-loss} = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]. \tag{5}$$

Essa métrica penaliza fortemente previsões confiantes e erradas. Já o Brier score mede o erro quadrático médio das probabilidades previstas:

$$\text{Brier} = \frac{1}{n}\sum_{i=1}^{n}(p_i - y_i)^2. \tag{6}$$

O uso conjunto dessas métricas é coerente com a avaliação de previsões probabilísticas discutida por Gneiting e Raftery (2007). A intenção não é apenas contar acertos, mas verificar se o modelo ordena melhor as partidas, evita probabilidades excessivamente confiantes e produz uma saída interpretável para uso pré-jogo.

2.5 Seleção de atributos

Como a base deste trabalho descreve cada confronto por meio de dezenas de atributos pré-jogo, nem toda variável disponível precisa entrar na configuração final do modelo. Quando muitos indicadores descrevem aspectos parecidos do mesmo fenômeno, o ajuste pode se tornar menos estável, menos interpretável e mais sensível ao conjunto de treinamento. Nesse contexto, selecionar atributos significa escolher quais sinais serão usados para representar cada confronto, sem criar informação nova.

A necessidade dessa etapa aparece porque várias estatísticas de Counter-Strike descrevem aspectos relacionados do desempenho das equipes. Indicadores como *rating*, *impact*, KAST, aberturas de *round*, razão entre abates e mortes e tempo

vivo por *round* podem carregar informação parcialmente sobreposta. Guyon e Elisseeff (2003) discutem a seleção de atributos como uma forma de reduzir ruído, melhorar a construção de modelos e facilitar a interpretação.

Também é importante distinguir seleção de atributos de seleção de modelo. A primeira decide quais variáveis entram na representação da partida; a segunda compara classificadores, hiperparâmetros e configurações de treinamento. Como as duas etapas envolvem escolhas entre alternativas, a geração e a comparação inicial de candidatos precisam ocorrer antes da janela reservada, que fica restrita à comparação final entre alternativas já selecionadas.

O principal risco metodológico, portanto, é obter uma estimativa otimista por escolher a alternativa que melhor se ajustou a uma amostra específica. Cawley e Talbot (2010) alertam que comparar muitas configurações pode gerar esse tipo de viés quando a avaliação final não é bem separada do processo de escolha. Por esse motivo, os ranqueamentos e comparações internas foram usados apenas dentro do conjunto de treinamento, enquanto a janela temporal de *holdout* foi preservada para a avaliação final das alternativas já selecionadas. Essa separação é compatível com Kohavi (1995), que trata a validação cruzada como estimativa interna de desempenho e apoio à seleção de modelos, não como substituto de uma avaliação final independente.

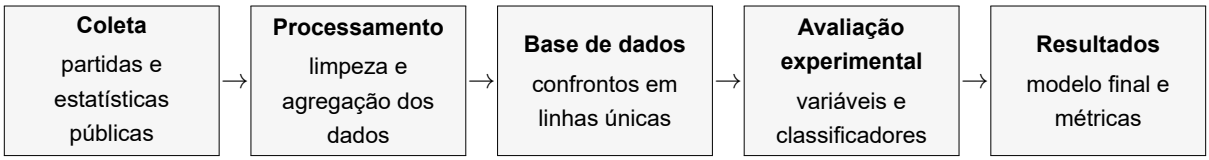
Na prática, essa lógica permite comparar subconjuntos de atributos sem transformar a escolha final em uma decisão manual orientada pelo resultado do *holdout*. Assim, a seleção de atributos cumpre dois papéis no trabalho: reduzir a redundância entre variáveis próximas e tornar mais defensável a comparação final entre configurações.

3 Metodologia

A construção da base foi organizada como um fluxo sequencial: coleta dos dados públicos, preparação dos confrontos, separação temporal, montagem dos atributos e seleção dos atributos usados na modelagem. Essa organização mantém a ligação entre os dados brutos e a tabela final usada pelos classificadores.

A Figura 2 apresenta a visão geral do fluxo metodológico. Cada bloco indica a função da etapa, mantendo a leitura centrada nos dados, nos atributos, na avaliação e nos resultados.

Figura 2 – Fluxo experimental em cinco blocos metodológicos.



Fonte: elaboração própria.

3.1 Base de dados e coleta

A base foi construída localmente a partir de dados públicos de partidas profissionais e estatísticas de jogadores. A coleta considerou dois grupos principais de informação: o histórico datado de confrontos e os indicadores agregados dos jogadores associados às equipes. As rotinas de coleta e processamento foram implementadas em Python, com apoio de bibliotecas como pandas e NumPy para leitura, limpeza, cruzamento e organização tabular dos dados.

A etapa de coleta automatizada foi realizada por *web scraping* em páginas públicas da HLTV.org. Como parte do conteúdo depende do carregamento da página no navegador, os scripts usaram Selenium com Chrome WebDriver em modo automatizado para abrir páginas de ranqueamento, páginas de equipes, páginas de estatísticas e resultados de partidas. Depois do carregamento, o HTML era analisado com BeautifulSoup, que permitia localizar blocos de times, jogadores, tabelas de estatísticas, datas e resultados. Para reduzir falhas, o processo repetia carregamentos quando necessário e verificava a integridade dos campos antes de gerar os arquivos tabulares.

O recorte de jogadores foi organizado por janelas sazonais (*seasons*). Neste trabalho, os rótulos s1 e s2 indicam, respectivamente, primeiro e segundo semestre do ano; a janela 2026_s1 é parcial, pois corresponde ao período reservado como *holdout*. Na abertura de cada janela, o ranqueamento da HLTV publicado naquele momento define as 50 equipes mais bem posicionadas e os jogadores associados a cada uma

delas. A escolha desse recorte é operacional, mas também dialoga com estudos que avaliam o ranqueamento da HLTV como fonte reconhecida e com poder explicativo para resultados em Counter-Strike (Rejthar; Kotrba, 2023). Para esses jogadores, foi criado um *snapshot*, isto é, um recorte estático com as estatísticas dos seis meses anteriores ao início da janela. Como o ranqueamento muda ao longo do tempo, os *snapshots* acumulam mais de 50 equipes distintas quando todas as janelas são observadas em conjunto. Em cada equipe e janela, os indicadores agregados foram calculados pela média dos jogadores disponíveis no *snapshot*, com limite operacional de até cinco jogadores por time. Esse desenho permite representar a força e o desempenho das equipes com informações disponíveis antes das partidas, sem misturar dados posteriores ao confronto.

Depois do cruzamento entre partidas, equipes, jogadores e *snapshots*, permaneceram 2.840 confrontos válidos para modelagem supervisionada. A Tabela 2 resume a composição usada no fluxo principal.

Tabela 2 – Resumo da base de dados usada no trabalho

Elemento	Descrição
Histórico bruto coletado	Confrontos datados entre 27/09/2023 e 22/03/2026; os registros anteriores a 2024_s1 funcionam como contexto prévio para sinais pré-jogo, não como exemplos supervisionados.
Recorte de equipes	Times associados às 50 equipes mais bem posicionadas em cada janela sazonal; ao longo das janelas, isso resulta em 97 equipes distintas nos *snapshots*.
Registros de jogadores	1.202 registros sazonais de jogadores nos *snapshots*.
Base supervisionada	2.840 partidas válidas após cruzamento entre partidas, equipes, jogadores e atributos pré-jogo.
Unidade de observação	Uma linha por partida, representada pela diferença entre a equipe analisada e a equipe adversária.

Fonte: elaboração própria.
Nota: Os valores resumem o recorte efetivamente usado no fluxo principal do trabalho.

3.2 Preparação dos confrontos e separação temporal

Depois da coleta, cada confronto foi transformado em uma única linha de dados. A mesma partida poderia ser escrita em duas orientações, por exemplo “equipe A contra equipe B” e “equipe B contra equipe A”, mas isso faria o mesmo confronto aparecer

duas vezes na base de dados. Por isso, a rotina de processamento manteve apenas uma representação, escolhida por uma regra fixa baseada nos nomes padronizados dos times. Essa regra serviu apenas para organizar a base: ela não depende de favoritismo, ELO ou resultado da partida. Assim, a equipe analisada é simplesmente o lado usado naquela linha, não necessariamente a favorita. A variável-alvo indica se essa equipe venceu a partida: valor 1 para vitória e valor 0 para derrota. As variáveis explicativas foram calculadas como diferenças entre os indicadores da equipe analisada e os indicadores do adversário. Portanto, um valor positivo representa vantagem relativa do time analisado, enquanto um valor negativo representa vantagem relativa do oponente.

O recorte prioriza consistência. Partidas em que não foi possível representar adequadamente as duas equipes com os dados disponíveis foram deixadas fora da base final. A filtragem reduz o tamanho da base, mas evita misturar exemplos com informação incompleta de elenco, histórico ou estatísticas agregadas.

A preparação também preservou a ordem temporal das partidas. Esse ponto é importante porque o objetivo é simular uma situação próxima ao uso real: modelos são ajustados com partidas antigas e avaliados em partidas posteriores. A Tabela 3 mostra a distribuição temporal da base supervisionada, isto é, das partidas que efetivamente viraram linhas de treinamento ou *holdout*. Embora o histórico bruto coletado comece em 27/09/2023, esses registros anteriores a 2024_s1 entram apenas como contexto prévio para cálculo de forma recente e ELO pré-jogo. O treinamento supervisionado começa em 2024_s1 e segue até 2025_s2; a janela 2026_s1 fica separada como *holdout* final.

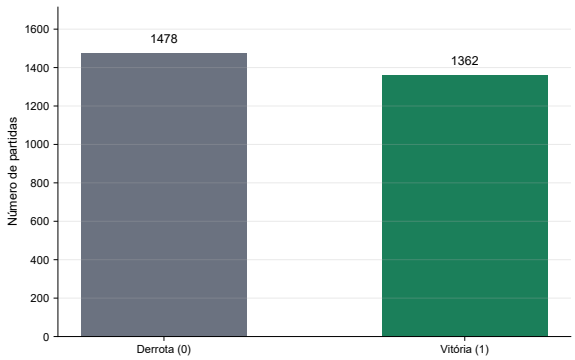
Tabela 3 – Distribuição temporal das partidas na base supervisionada

Janela	Partidas
2024_s1	574
2024_s2	670
2025_s1	688
2025_s2	614
2026_s1	294

Fonte: elaboração própria.

Nota: As quatro primeiras janelas compõem o conjunto de treinamento e seleção; a janela 2026_s1 foi preservada como *holdout* final.

Figura 3 – Distribuição das classes na base utilizada: 1.478 derrotas e 1.362 vitórias para a equipe analisada.



Fonte: elaboração própria.

A Figura 3 mostra que a base não é dominada por uma única classe. São 1.478 derrotas e 1.362 vitórias para a equipe analisada, uma diferença pequena o suficiente para evitar que a avaliação seja explicada apenas por uma classe majoritária.

3.3 Construção dos atributos

A maior parte dos atributos corresponde a diferenças entre os times, pois uma partida é uma comparação direta. Em vez de representar uma equipe de modo isolado, a linha compara o time analisado com o adversário na mesma data. Essa representação se aproxima de problemas de comparação pareada, em que dois objetos são comparados e um resultado é observado (Cattelan, 2012).

Os atributos combinam sinais de força competitiva, forma recente e desempenho agregado dos jogadores. Entre eles estão a diferença de ELO pré-jogo, a taxa recente de vitórias, calculada sobre as 30 partidas anteriores ao confronto, e estatísticas médias dos jogadores disponíveis no *snapshot* da equipe, como *rating*, KAST, *impact*, razão entre abates e mortes, mortes por *round*, tempo vivo por *round* e indicadores de abertura. Como o KAST resume participação ou sobrevivência em rodadas, ele complementa medidas mais diretas de abates e mortes. Essa organização busca representar a vantagem relativa entre as duas equipes antes do confronto, sem usar informação observada durante ou depois da partida. A lista completa dos 38 atributos considerados na bateria experimental é apresentada no Apêndice A.

O ELO tem papel específico nessa representação porque resume a força competitiva acumulada de cada equipe ao longo do histórico de partidas. Diferentemente de uma estatística individual de jogador, ele tenta capturar a trajetória do time como unidade competitiva. Por isso, a diferença de ELO pré-jogo funciona como uma medida sintética de vantagem histórica antes do confronto, uso semelhante ao avaliado

por Hvattum e Arntzen na predição de resultados no futebol profissional (Hvattum; Arntzen, 2010).

O ELO foi atualizado cronologicamente. Para uma partida entre os times A e B, a expectativa de vitória de A é:

$$E_A = \frac{1}{1 + 10^{(R_B - R_A)/400}}. \tag{7}$$

Depois da partida, a pontuação é atualizada por:

$$R'_A = R_A + 32(S_A - E_A), \tag{8}$$

em que R_A é a pontuação anterior, S_A vale 1 em caso de vitória e 0 em caso de derrota, e 32 é o fator K adotado nesta implementação. Em termos práticos, K controla o tamanho da atualização: quanto maior esse valor, mais cada resultado altera a pontuação da equipe. Todas as equipes iniciam o histórico com 1500 pontos, e a escala da expectativa é 400, valor que converte a diferença entre pontuações em expectativa de vitória. Em cada linha, o valor registrado corresponde à pontuação antes da partida, impedindo que o resultado do confronto influencie a variável usada para prevê-lo. O uso de um fator K fixo é uma simplificação: a pontuação ELO não diferencia formato da série, importância do campeonato ou quantidade de mapas disputados.

3.4 Seleção dos atributos

A etapa seguinte definiu quais subconjuntos de atributos seriam levados ao treinamento. Essa seleção ocorreu antes do *holdout*, por regras fixas de ranqueamento e blocos de informação. O objetivo foi reduzir redundância, pois várias estatísticas de Counter-Strike medem fenômenos próximos, como desempenho individual, impacto em aberturas e sobrevivência por *round*.

A seleção de atributos foi usada para reduzir redundância e manter subconjuntos mais interpretáveis, em linha com Guyon e Elisseeff (2003). Com essa função, a bateria experimental organizou cinco estratégias próprias de comparação: ranqueamento univariado, seleção associada à regressão logística L1, importância em modelos baseados em árvores (Breiman, 2001), consenso entre ranqueamentos e blocos compactos de variáveis. No ranqueamento univariado, cada atributo foi testado sozinho em divisões internas do treinamento, usando uma regressão logística simples para medir seu ROC-AUC médio; log-loss, Brier score e acurácia foram usados como critérios de desempate. O rótulo *univariate top 11* se refere às 11 variáveis mais bem posicionadas nesse filtro; não significa que o modelo final use apenas uma variável. Também foram avaliados refinos locais, sem enumerar todas as combinações possíveis.

4 Avaliação experimental e resultados

A avaliação experimental comparou os classificadores sob o mesmo recorte de dados, o mesmo pré-processamento e a mesma janela posterior de comparação. Como os melhores resultados ficaram próximos, a escolha final também considerou estabilidade interna, interpretação e reprodutibilidade.

4.1 Protocolo de avaliação

A Tabela 4 resume o desenho usado para separar treino, validação interna e comparação final dos modelos.

Tabela 4 – Resumo do protocolo experimental

Item	Definição usada no experimento
Unidade de análise	Uma partida profissional, representada por diferenças entre a equipe analisada e a adversária.
Treinamento, validação interna e seleção	Janelas 2024_s1, 2024_s2, 2025_s1 e 2025_s2, totalizando 2.546 partidas.
Comparação posterior	Janela 2026_s1, mantida fora do ajuste dos modelos e usada para comparar alternativas já selecionadas, com 294 partidas.
Atributos de entrada	38 atributos pré-jogo construídos a partir de força histórica, forma recente e estatísticas agregadas dos jogadores, além de indicadores de contexto da partida e de disponibilidade de elenco.
Espaço comparado	71 conjuntos de variáveis, 58 configurações possíveis de classificadores e 2.582 combinações efetivamente avaliadas.
Pré-processamento	Normalização *z-score* dentro do fluxo de treinamento para modelos sensíveis à escala.
Métricas	ROC-AUC, acurácia, precisão, revocação, F1-score, log-loss e Brier score.

Fonte: elaboração própria.
Nota: O protocolo resume a seleção inicial, a comparação em janela posterior e a escolha da configuração final.

A avaliação usa as janelas de 2024_s1 a 2025_s2 para treinamento e escolha preliminar dos candidatos, preservando 2026_s1 como *holdout*, isto é, a janela final de dados reservada apenas para a avaliação, sem participar do treinamento nem da

seleção interna. Assim, os parâmetros são estimados com partidas anteriores, e a janela final compara as alternativas finalistas sem reabrir o processo de escolha.

Dentro do conjunto de treinamento, foram usadas validações internas com funções diferentes. Na seleção de atributos, os ranqueamentos foram calculados em divisões internas do próprio treino; quando havia janelas suficientes, essas divisões respeitavam a ordem temporal das *seasons*. Para os modelos finalistas, a métrica CV ROC-AUC, reportada adiante na Tabela 6, corresponde à validação cruzada estratificada repetida, com cinco *folds* e duas repetições. Em termos práticos, isso significa que o conjunto de treinamento foi dividido em cinco partes: em cada rodada, quatro partes foram usadas para treinar e uma para validar, até que todas fossem usadas como validação. Esse procedimento foi repetido duas vezes para reduzir a dependência de uma única divisão dos dados. Como essa divisão estratificada preserva a proporção de vitórias e derrotas, mas não preserva necessariamente a ordem cronológica entre os *folds*, seus valores foram tratados apenas como diagnóstico interno de estabilidade, e não como evidência temporal de generalização.

A comparação final entre alternativas já selecionadas foi feita no *holdout*. Modelos sensíveis à escala, como KNN, SVM e regressão logística, usaram normalização *z-score* dentro do próprio fluxo de treinamento, colocando as variáveis em escala comum sem usar informação da janela de teste.

A execução da bateria experimental foi feita localmente em CPU, usando um processador Ryzen 9 7900, com 12 núcleos e 24 threads. O tempo de execução não foi usado como critério de comparação entre classificadores; a avaliação priorizou métricas preditivas e probabilísticas. Os modelos tabulares foram implementados em scikit-learn, com pré-processamento, treinamento, validação e cálculo de métricas no mesmo ambiente.

4.2 Modelos avaliados e espaço de busca

Para comparar os classificadores sob condições equivalentes, foram combinados subconjuntos de variáveis e configurações de modelos em uma grade controlada. Os conjuntos vieram de ranqueamentos univariados, seleção L1, importância por árvores, consenso entre ranqueamentos, blocos compactos e refinos locais. As configurações reuniram variações dos classificadores testados, como diferentes valores de regularização, pesos de classe, número de vizinhos, suavização do Bayes gaussiano e ajustes de SVM e *Random Forest*. O total de 2.582 combinações vem de duas fases: na primeira, foram avaliados 39 subconjuntos candidatos, todos formados a partir dos mesmos 38 atributos pré-jogo, em combinação com 58 configurações de classificadores, totalizando 2.262 candidatos; na segunda, 32 subconjuntos de refino foram

avaliados com 10 configurações, somando mais 320 candidatos. A Tabela 5 resume essa grade.

Tabela 5 – Classificadores e variações avaliadas

Classificador	O que foi variado
Regressão logística	Regularização L2, L1 e *elastic net*; diferentes intensidades de regularização; versões com e sem balanceamento de classes.
Bayes gaussiano	Suavização da variância em diferentes ordens de grandeza, mantendo o classificador probabilístico simples.
KNN	Número de vizinhos $k = 1$, 5 e 10, com pesos uniformes ou por distância, sempre após normalização *z-score*.
SVM	Separadores linear e RBF, com variação de regularização e parâmetros do núcleo não linear.
Random Forest	Quantidade de árvores, profundidade máxima, mínimo de amostras por folha e número de atributos considerados por divisão.
Referência simples	Regra majoritária e configurações básicas usadas como piso de comparação.

Fonte: elaboração própria.
Nota: A tabela apresenta a lógica geral da grade experimental. As 58 configurações resumem as variações de classificadores; elas foram avaliadas em combinações controladas com os conjuntos de atributos.

Esse desenho permitiu cobrir alternativas lineares, probabilísticas, KNN, SVM e árvores.

A busca não enumerou todos os subconjuntos possíveis das 38 variáveis. Essa decisão foi metodológica: uma enumeração completa criaria um espaço grande demais para um *holdout* de 294 partidas e aumentaria o risco de selecionar uma combinação favorecida por variação amostral. Em vez disso, foram comparadas famílias de subconjuntos geradas por critérios explícitos, como ranqueamento univariado, estabilidade L1, importância em árvores e combinações por blocos de informação.

4.3 Resultado principal e escolha do modelo

Após a seleção interna, os melhores representantes por classificador foram comparados no *holdout* temporal. A Tabela 6 apresenta essa comparação e destaca o modelo final escolhido. A decisão não depende apenas do maior valor isolado em uma métrica, mas do equilíbrio entre desempenho, interpretação e reprodutibilidade. Como a busca combinou cada classificador com diferentes subconjuntos de atributos, o número de variáveis não é fixo em todas as linhas. A coluna Var. indica o tamanho do subconjunto associado ao melhor representante de cada classificador.

Tabela 6 – Comparação dos melhores representantes por classificador

Classificador	Var.	ROC-AUC	Acurácia	CV ROC-AUC
Logística L2 (final)	11	**0,6828**	**0,6293**	**0,6485**
SVM linear	11	0,6828	0,6259	0,6480
Logística L1	11	0,6823	0,6190	0,6485
Logística *elastic net*	8	0,6811	0,6293	0,6486
Bayes gaussiano	8	0,6762	0,6224	0,6436
SVM RBF	11	0,6650	0,6224	0,6362
Random Forest	20	0,6625	0,6054	0,6075
KNN	11	0,6255	0,6054	0,5947

Fonte: elaboração própria.
Nota: CV ROC-AUC corresponde à média da validação cruzada estratificada repetida dentro do treino, com cinco *folds* e duas repetições. A comparação temporal final foi feita no *holdout* 2026_s1.

A regressão logística L2 e o SVM linear ficaram empatados em ROC-AUC no *holdout*. Essa situação foi tratada como empate prático, não como evidência de superioridade estatística de um classificador sobre o outro. A regressão logística foi mantida por razões operacionais claras: apresentou acurácia ligeiramente maior no limiar padrão de 0,5, produz probabilidades diretamente interpretáveis e permite leitura por coeficientes.

Diante da diferença pequena entre os dois modelos, não havia ganho suficiente para trocar a solução por um classificador menos transparente. As variações L1 e *elastic net* também ficaram próximas, o que indica que a regressão logística foi competitiva sob diferentes penalizações. Bayes gaussiano, *Random Forest*, SVM RBF e KNN ficaram abaixo nessa comparação final.

Como verificação complementar, também foi treinada uma regressão logística usando apenas a diferença de ELO pré-jogo. Esse modelo de referência obteve ROC-AUC de 0,6686 e acurácia de 0,6327 no *holdout* 2026_s1, com log-loss de 0,6451 e Brier score de 0,2274. O resultado indica que o ELO é um sinal individual forte, mas

o modelo final com 11 variáveis apresentou melhor ordenação probabilística e melhor qualidade das probabilidades previstas. Assim, as demais variáveis acrescentam informação complementar, embora o ganho sobre ELO isolado seja moderado.

O modelo final foi a configuração *univariate top 11* com regressão logística L2, $C = 0{,}3$, sem balanceamento artificial de classes e algoritmo de otimização `lbfgs`. No *holdout* 2026_s1, essa configuração obteve ROC-AUC de 0,6828 e acurácia de 0,6293. A versão equivalente com balanceamento de classes empatou em ROC-AUC e acurácia, mas alterou o perfil de erros: aumentou revocação e F1-score, ao custo de piora leve em log-loss e Brier score. Como a base é relativamente equilibrada, a versão sem balanceamento foi mantida por simplicidade e por não introduzir correção adicional sem ganho claro nas métricas centrais.

Tabela 7 – Configuração final do modelo

Item	Configuração final
Conjunto de variáveis	11 atributos selecionados por ranqueamento univariado no treino.
Classificador	Regressão logística com regularização L2.
Hiperparâmetros	$C = 0{,}3$, `class_weight=None`, algoritmo de otimização `lbfgs`.
Pré-processamento	Normalização *z-score* dentro do fluxo de treinamento.
Limiar de decisão	0,5 para cálculo das métricas binárias principais.
Avaliação final	*holdout* 2026_s1, com 294 partidas.

Fonte: elaboração própria.
Nota: O modelo final foi definido antes das análises de robustez e da avaliação prospectiva complementar. As métricas binárias principais foram calculadas com o limiar padrão 0,5 para manter uma regra de decisão simples e comparável.

A classe majoritária do *holdout* teria acurácia de 0,5238; portanto, o modelo final acrescenta cerca de 10,5 pontos percentuais em relação a essa regra trivial. O resultado não deve ser lido como previsão determinística. Ele mostra um sinal probabilístico moderado, compatível com um domínio em que mapas, vetos, momento individual e decisões táticas ainda podem alterar o resultado de uma partida.

A escolha final combina desempenho observado, clareza de interpretação e reprodutibilidade. Como os ganhos entre classificadores foram moderados, a regressão logística L2 foi preferida por entregar resultado competitivo com 11 variáveis e uma leitura mais transparente do papel dos atributos, sem exigir núcleo não linear ou ajuste adicional de pesos de classe.

4.4 Desempenho do modelo final

Com a configuração final definida, a análise seguinte detalha o desempenho no *holdout* 2026_s1 por métricas complementares e pela curva ROC. O objetivo é mostrar como o resultado se comporta na janela final, sem reabrir a escolha do classificador.

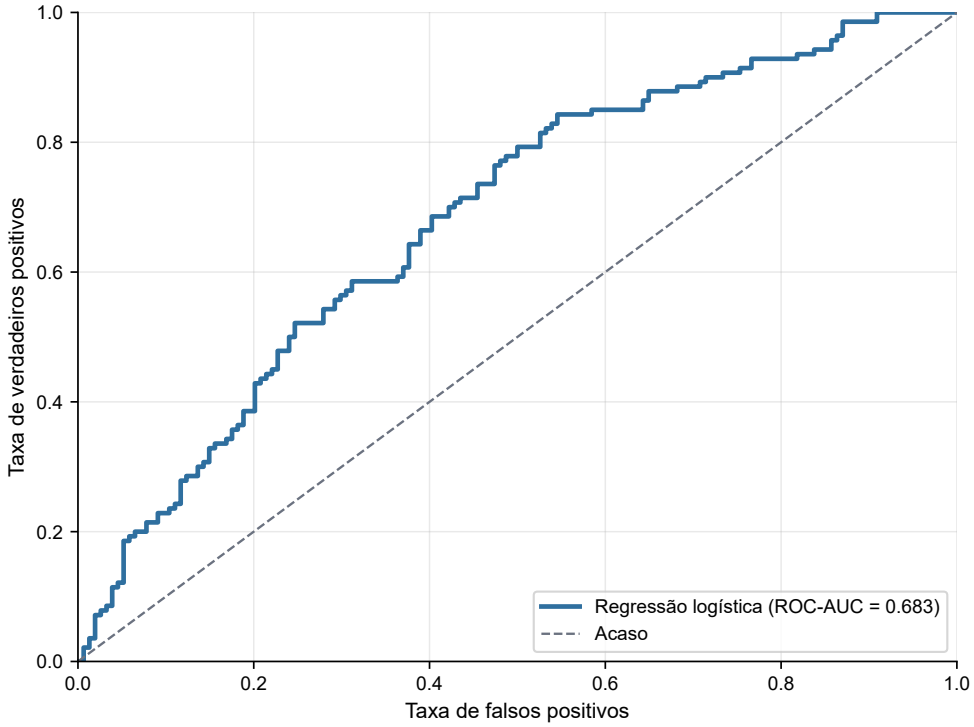
Tabela 8 – Métricas do modelo final no *holdout* 2026_s1

Métrica	Valor
ROC-AUC	0,6828
Acurácia	0,6293
Precisão	0,6165
Revocação	0,5857
F1-score	0,6007
Log-loss	0,6388
Brier score	0,2243

Fonte: elaboração própria.
Nota: O *holdout* contém 294 partidas da janela 2026_s1. Para ROC-AUC, acurácia, precisão, revocação e F1-score, valores maiores são desejáveis; para log-loss e Brier score, valores menores indicam melhor qualidade probabilística.

A Tabela 8 mostra duas leituras complementares. A acurácia descreve a decisão binária quando se aplica o limiar 0,5. Já ROC-AUC, log-loss e Brier score avaliam a qualidade da ordenação e das probabilidades com mais nuance. Essa separação é importante porque o modelo foi pensado para gerar probabilidades pré-jogo, e não apenas uma etiqueta de vencedor. Em termos práticos, o resultado indica um classificador útil para ordenar graus de favoritismo, mas ainda distante de um sistema de alta certeza partida a partida.

Figura 4 – Curva ROC do modelo final no *holdout* 2026_s1.



Fonte: elaboração própria.

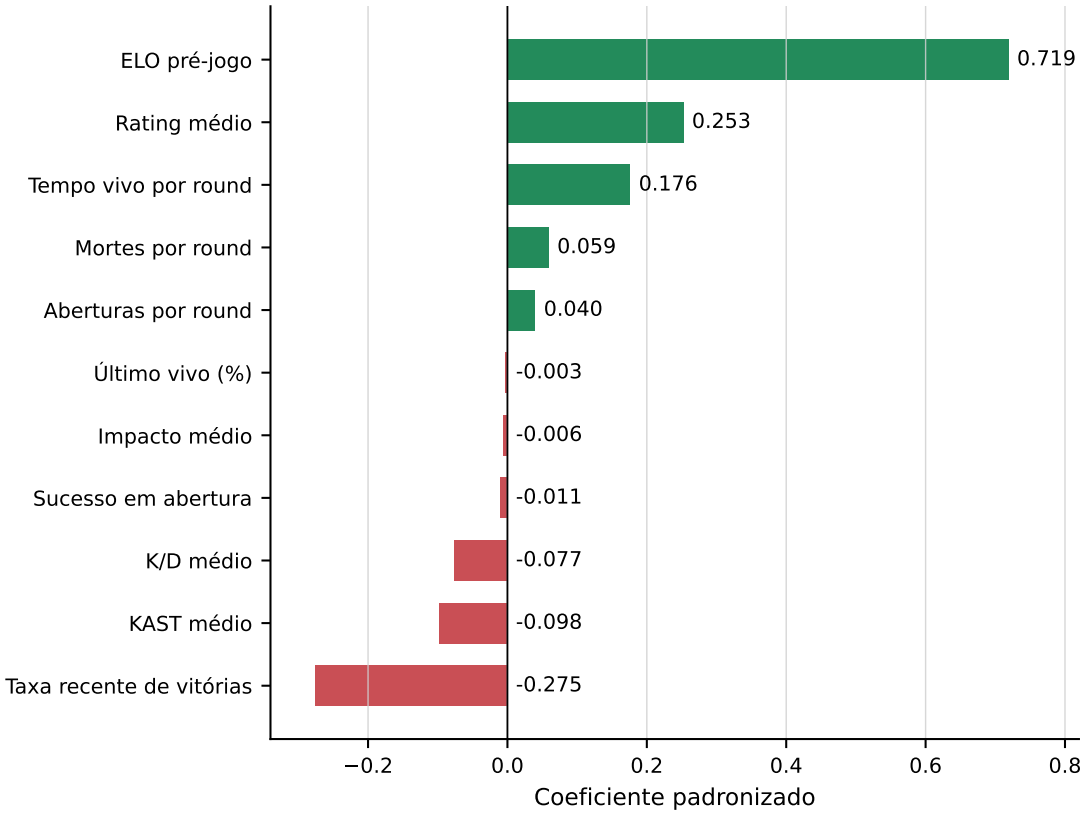
A Figura 4 apresenta a curva ROC da regressão logística final no *holdout*. O eixo horizontal mostra a taxa de falsos positivos e o eixo vertical mostra a taxa de verdadeiros positivos quando o limiar de decisão é deslocado de valores mais conservadores para valores mais permissivos. A diagonal representa o comportamento esperado de uma escolha aleatória; por isso, quanto mais a curva se afasta dessa diagonal em direção ao canto superior esquerdo, melhor é a capacidade de ordenar partidas vencidas acima de partidas perdidas. O ROC-AUC de 0,6828 resume essa área sob a curva e pode ser lido como um sinal preditivo moderado: o modelo usa informação real dos atributos pré-jogo, mas ainda opera em um domínio no qual o resultado de uma partida profissional depende de fatores que não aparecem completamente na base.

No limiar padrão de 0,5, o modelo acertou 185 das 294 partidas do *holdout*. Foram 103 verdadeiros negativos e 82 verdadeiros positivos. Os erros também apareceram nos dois sentidos: 51 derrotas foram previstas como vitórias e 58 vitórias foram previstas como derrotas. Essa distribuição reforça que o classificador deve ser lido como uma ferramenta probabilística, especialmente útil para indicar vantagem relativa, e não como mecanismo determinístico de acerto.

4.5 Coeficientes e leitura das variáveis

A regressão logística permite examinar os pesos associados às variáveis, desde que essa leitura seja feita com cautela. Como os atributos são padronizados por *z-score* dentro do fluxo de treinamento, os coeficientes ficam em escala comparável. Valores positivos aumentam a probabilidade estimada de vitória do time analisado; valores negativos reduzem essa probabilidade, considerando o conjunto de variáveis no mesmo modelo.

Figura 5 – Coeficientes padronizados das diferenças entre equipes no modelo logístico final



Fonte: elaboração própria.

A Figura 5 mostra que o ELO pré-jogo, sempre calculado como diferença entre as equipes, é o sinal individual de maior peso positivo. Em seguida aparecem indicadores ligados a desempenho médio dos jogadores, como *rating* e tempo vivo por *round*. Essa leitura é coerente com a formulação do problema: a previsão compara duas equipes antes da partida e tende a valorizar sinais que resumem força competitiva e desempenho agregado.

Os coeficientes negativos exigem uma leitura mais cuidadosa. A taxa recente de vitórias, por exemplo, é um sinal útil isoladamente, mas também acompanha a força histórica do time: no conjunto de treinamento, sua correlação com a diferença de ELO foi alta. Quando essas variáveis entram juntas na regressão logística, o peso da taxa

recente passa a representar apenas a contribuição que sobra depois de controlar ELO, *rating*, KAST e outros indicadores. Por isso, um coeficiente negativo não significa que vencer recentemente seja ruim; indica apenas como o modelo redistribui informação entre variáveis parecidas. Ainda assim, os coeficientes tornam a decisão mais transparente e ajudam a justificar o uso de um modelo interpretável.

4.6 Avaliação prospectiva em eventos posteriores

Após a definição do modelo final, foi feita uma avaliação prospectiva complementar em três eventos posteriores de maio de 2026: PGL Astana, disputado entre 09/05 e 17/05; IEM Atlanta, entre 11/05 e 17/05; e CS Asia Championships, entre 20/05 e 24/05. Em cada confronto, a configuração final permaneceu fixa, sem nova seleção de variáveis, ajuste de hiperparâmetros ou troca de classificador. O objetivo foi observar como o modelo já treinado se comporta em partidas acompanhadas posteriormente, usando estatísticas disponíveis antes dos jogos. As previsões foram registradas em planilhas de acompanhamento antes da consolidação final dos resultados; por isso, essa etapa deve ser entendida como verificação operacional complementar. A Tabela 9 reporta os resultados obtidos com o *snapshot* operacional de 12 meses, calculado com estatísticas dos jogadores entre 02/05/2025 e 02/05/2026, inclusive, e congelado em 02/05/2026 como leitura principal dessa etapa.

Tabela 9 – Avaliação prospectiva complementar com *snapshot* de 12 meses

Evento	Jogos	Acertos	Acurácia	ROC-AUC
PGL Astana 2026	41	30/41	0,7317	0,8206
IEM Atlanta 2026	30	20/30	0,6667	0,7009
CS Asia Championships 2026	30	19/30	0,6333	0,7060
Total	**101**	**69/101**	**0,6832**	**0,7304**

Fonte: elaboração própria.

Nota: Modelo já definido e estatísticas disponíveis antes dos jogos acompanhados. A tabela apresenta o *snapshot* operacional de 12 meses, adotado como leitura principal da avaliação complementar.

A Tabela 9 acrescenta uma verificação operacional em partidas posteriores ao *holdout*, mas não substitui a comparação final feita na janela 2026_s1. A diferença de desempenho entre a prospectiva e o *holdout* também não deve ser lida como melhora direta do modelo: os eventos acompanhados reuniram mais partidas com favoritos claros, com mediana de diferença absoluta de ELO de 141,6 pontos, contra 78,2 pontos no *holdout* 2026_s1. Uma hipótese plausível é que essa maior separação pré-jogo tenha favorecido o desempenho prospectivo, já que o modelo tende a se comportar melhor quando existe diferença clara entre as equipes.

Ainda assim, em 101 partidas acompanhadas posteriormente, o modelo manteve ROC-AUC acima de 0,70 nos três eventos e acurácia agregada de 0,6832. Como cada torneio possui amostra pequena, os valores por evento devem ser lidos como descrição do acompanhamento, não como estimativas estatísticas definitivas. As previsões em que o modelo indicou maior confiança também foram favoráveis, mas representam um subconjunto pequeno; o acompanhamento registrou um erro de alta confiança em BetBoom contra Vitality na IEM Atlanta 2026, o que reforça a necessidade de cautela.

4.7 Análises complementares de robustez

A primeira verificação de robustez observa se o modelo final mantém desempenho em janelas temporais sucessivas da própria base. Nessa validação, não houve nova seleção de variáveis, nova busca de hiperparâmetros ou mudança de classificador. O treino cresce ao longo do tempo e a janela seguinte é usada como teste.

Tabela 10 – Validação temporal bloqueada do modelo final

Janela de teste	n treino	n teste	Baseline	ROC-AUC	Acurácia
2024_s2	574	670	0,5478	0,6456	0,6015
2025_s1	1244	688	0,5000	0,6476	0,6003
2025_s2	1932	614	0,5228	0,6695	0,6026
2026_s1	2546	294	0,5238	0,6828	0,6293

Fonte: elaboração própria.
Nota: A validação usa o mesmo modelo final, sem nova seleção de variáveis ou hiperparâmetros.

A Tabela 10 mostra que o ROC-AUC permaneceu acima de 0,64 nas quatro janelas e foi maior na última delas, correspondente a 2026_s1. A acurácia ficou próxima de 0,60 nas três primeiras janelas e chegou a 0,6293 em 2026_s1. Essa validação sugere estabilidade temporal retrospectiva dentro da base construída para o trabalho. A diferença entre janelas, porém, deve ser lida com cuidado, pois pode refletir mudanças no recorte competitivo, aumento da quantidade de dados de treino e composição específica de cada semestre.

4.8 Variação por contexto

Além da estabilidade temporal, os recortes por contexto mostram em quais tipos de partida o modelo foi mais forte ou mais fraco. Esses recortes são diagnósticos do *holdout* e não alteram a configuração final.

Tabela 11 – Desempenho do modelo principal por contexto no *holdout*

Recorte	n	ROC-AUC	Acurácia
Partidas *online*	121	0,6104	0,5620
Partidas LAN	173	0,7332	0,6763
Menor quartil de confiança	74	0,5084	0,5000
Maior quartil de confiança	74	0,7517	0,7432
Menor quartil de diferença absoluta de ELO	74	0,5241	0,5270
Maior quartil de diferença absoluta de ELO	74	0,7693	0,7297

Fonte: elaboração própria.
Nota: Os recortes são diagnósticos do *holdout*. Eles mostram variação de desempenho por contexto, mas não substituem a avaliação principal.

A Tabela 11 indica que o desempenho observado foi maior em partidas LAN, em previsões de maior confiança e em confrontos com maior diferença absoluta de ELO. Em contrapartida, quando as equipes estão muito próximas ou quando a probabilidade estimada fica perto de 50%, o desempenho se aproxima do acaso. A diferença entre LAN e *online* não deve ser lida como prova causal, mas é coerente com a ideia de que formato, premiação, estrutura do torneio e ambiente competitivo podem afetar o desempenho observado em Counter-Strike (Shenkman; Molodchik; Parshakov, 2025). Essa leitura delimita o alcance do modelo: a contribuição não é afirmar que todas as partidas são previsíveis, mas mostrar em quais situações o sinal pré-jogo parece mais informativo.

5 Discussão

O desempenho observado é moderado e contextual. O modelo supera a regra majoritária e mantém ROC-AUC acima do acaso, mas funciona melhor quando existe diferença clara entre as equipes e perde poder preditivo em confrontos equilibrados. Por isso, a leitura mais adequada não é a de um classificador capaz de resolver qualquer confronto, mas a de um fluxo que identifica vantagem relativa quando os sinais pré-jogo são suficientemente claros.

Em um conjunto amplo de comparações, diferenças pequenas podem refletir sensibilidade da amostra, problema discutido na literatura de seleção de modelos (Cawley; Talbot, 2010). Por isso, a decisão final não dependeu apenas do maior valor isolado em uma métrica, mas da combinação entre desempenho observado, transparência e reprodução do experimento. A contribuição do trabalho está nesse fechamento: construir um fluxo rastreável, comparar alternativas sob o mesmo protocolo e escolher uma configuração cujo comportamento possa ser explicado.

5.1 Colinearidade e leitura dos coeficientes

Os coeficientes da regressão logística ajudam a entender como o modelo organiza as variáveis, mas não devem ser lidos como efeitos isolados. No conjunto final, algumas estatísticas descrevem aspectos próximos do desempenho dos jogadores. Por isso, é esperado que variáveis como *rating*, razão entre abates e mortes, *impact* e KAST carreguem parte da mesma informação.

Essa proximidade apareceu nas correlações entre atributos. As maiores correlações absolutas foram observadas entre *rating* médio e razão entre abates e mortes, 0,9234; *rating* médio e *impact*, 0,9017; e razão entre abates e mortes e KAST, 0,8711. Em termos práticos, isso significa que essas variáveis tendem a variar juntas: quando uma equipe aparece melhor em uma delas, muitas vezes também aparece melhor nas outras.

Essa colinearidade não impede o uso do modelo, mas limita a leitura individual dos pesos. A regularização L2 ajuda a distribuir os pesos entre variáveis relacionadas, sem eliminar automaticamente uma delas. Por isso, a visualização dos coeficientes é útil para entender a direção geral da decisão, mas não deve ser lida como uma decomposição causal do resultado da partida. Em um modelo multivariado, o coeficiente de uma variável representa sua contribuição controlando as demais variáveis; pesos negativos em alguns indicadores mostram como o modelo redistribui informação quando sinais semelhantes entram juntos.

5.2 Limitações e continuidade

As principais limitações decorrem do nível de detalhe dos dados, do tamanho da janela final e do risco de seleção após muitas comparações. O *holdout* contém 294 partidas, quantidade adequada para uma avaliação inicial, mas ainda restrita para conclusões fortes sobre todos os contextos competitivos. Os atributos agregados nos *snapshots* aproximam força, forma recente e desempenho, mas não reconstroem perfeitamente escalação, mapas, vetos, lado inicial, viagem, preparação ou mudanças táticas; esses elementos não entram como variáveis do modelo final.

O modelo também não foi comparado com *odds* de mercado nem passou por calibração probabilística, dois caminhos úteis para avaliar uso prático das probabilidades em trabalhos futuros. A separação temporal, a validação cruzada interna e a comparação final no *holdout* reduzem o risco de viés de seleção, mas não o eliminam completamente. Também não foram aplicados testes formais para diferença entre ROC-AUCs, de modo que empates e diferenças pequenas foram tratados com cautela metodológica. Por isso, as conclusões valem para o recorte estudado: equipes do topo do cenário, partidas representáveis pelos dados disponíveis e janelas históricas acompanhadas neste trabalho.

Como continuidade, a linha temporal merece investigação adicional. Experimentos complementares com variáveis temporais indicaram que informações de forma recente, força dos adversários e contexto competitivo podem acrescentar sinal preditivo, mas essa linha exigiria reconstruir essas variáveis antes de cada nova partida, com atualização contínua de partidas recentes, elencos, estatísticas e contexto LAN/*online*. Por esse motivo, ela fica como continuidade metodológica, enquanto o modelo principal permanece como configuração mais simples e reprodutível.

Em síntese, as análises complementares delimitam o uso do modelo: ele é mais informativo quando existe diferença clara entre as equipes e menos confiável em confrontos equilibrados. O trabalho não promete prever qualquer partida; organiza um fluxo que identifica sinal pré-jogo moderado, interpretável e verificável.

6 Conclusão

Este trabalho apresentou um fluxo reprodutível para predição pré-jogo de partidas profissionais de Counter-Strike 2. A proposta combinou coleta de dados públicos, construção de uma base de dados por diferenças entre equipes, seleção de atributos, comparação de classificadores e avaliação em ordem temporal. Com isso, a contribuição do trabalho está menos em um algoritmo isolado e mais em conectar as etapas experimentais de forma que possam ser conferidas e reproduzidas.

A configuração final foi uma regressão logística L2 com 11 variáveis, $C = 0{,}3$, sem balanceamento artificial de classes e com normalização *z-score* no fluxo de treinamento. No *holdout* 2026_s1, o modelo atingiu ROC-AUC de 0,6828 e acurácia de 0,6293, superando a regra de classe majoritária. O SVM linear teve desempenho muito próximo em ROC-AUC, mas a regressão logística foi mantida por oferecer probabilidade direta, coeficientes interpretáveis, acurácia ligeiramente maior no limiar padrão e maior reprodutibilidade do experimento.

As análises complementares ajudam a delimitar o alcance do modelo. O desempenho foi melhor em partidas LAN, em previsões de maior confiança e em confrontos com maior diferença de ELO, enquanto partidas equilibradas continuam sendo mais difíceis. A avaliação prospectiva em eventos posteriores reforçou essa leitura como verificação operacional. Assim, a conclusão central é moderada: há sinal pré-jogo útil no recorte estudado, mas esse sinal depende do contexto competitivo e não deve ser interpretado como garantia de acerto em partidas individuais.

Como trabalhos futuros, recomenda-se aprofundar a construção operacional de variáveis temporais, incorporar informações de mapas e vetos, registrar mudanças de elenco com maior precisão, comparar o modelo com *odds* de mercado registradas antes das partidas e ampliar a avaliação prospectiva em novos eventos. Esses caminhos ampliam o escopo do trabalho sem alterar a configuração final avaliada aqui.

REFERÊNCIAS

BJÖRKLUND, Arvid; LINDEVALL, Fredrik; SVENSSON, Philip; VISURI, William Johansson. *Predicting the outcome of CS:GO games using machine learning*. Trabalho de conclusão de curso (Bachelor of Science Thesis), Chalmers University of Technology, Gothenburg, 2018.

BREIMAN, Leo. Random Forests. *Machine Learning*, v. 45, p. 5–32, 2001.

BROMS, Erik; NORDANSJÖ, William. *Predicting Counter-Strike Matches Using Machine Learning Models*. Trabalho de conclusão de curso em Estatística, Lund University School of Economics and Management, 2024.

CATTELAN, Manuela. Models for Paired Comparison Data: A Review with Emphasis on Dependent Data. *Statistical Science*, v. 27, n. 3, 2012. DOI: 10.1214/12-STS396.

CAWLEY, Gavin C.; TALBOT, Nicola L. C. On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *Journal of Machine Learning Research*, v. 11, p. 2079–2107, 2010.

CORTES, Corinna; VAPNIK, Vladimir. Support-vector networks. *Machine Learning*, v. 20, n. 3, p. 273–297, 1995. DOI: 10.1007/BF00994018.

COVER, Thomas M.; HART, Peter E. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, v. 13, n. 1, p. 21–27, 1967. DOI: 10.1109/TIT.1967.1053964.

ELO, Arpad E. *The Rating of Chessplayers, Past and Present*. New York: Arco Publishing, 1978.

FAWCETT, Tom. An introduction to ROC analysis. *Pattern Recognition Letters*, v. 27, n. 8, p. 861–874, 2006.

GNEITING, Tilmann; RAFTERY, Adrian E. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, v. 102, n. 477, p. 359–378, 2007. DOI: 10.1198/016214506000001437.

GUYON, Isabelle; ELISSEEFF, André. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, v. 3, p. 1157–1182, 2003.

HAND, David J.; YU, Keming. Idiot's Bayes — Not So Stupid After All? *International Statistical Review*, v. 69, n. 3, p. 385–398, 2001. DOI: 10.1111/j.1751-5823.2001.tb00465.x.

HODGE, Victoria J.; DEVLIN, Sam Michael; SEPHTON, Nicholas John; BLOCK, Florian Oliver; COWLING, Peter Ivan; DRACHEN, Anders. *Win Prediction in Multi-Player Esports: Live Professional Match Prediction*. IEEE Transactions on Games, v. 13, n. 4, p. 368–379, 2021. DOI: 10.1109/TG.2019.2948469.

HOSMER, David W.; LEMESHOW, Stanley; STURDIVANT, Rodney X. *Applied Logistic Regression*. 3. ed. Hoboken: Wiley, 2013.

HVATTUM, Lars Magnus; ARNTZEN, Halvard. Using ELO ratings for match result prediction in association football. *International Journal of Forecasting*, v. 26, n. 3, p. 460–470, 2010. DOI: 10.1016/j.ijforecast.2009.10.002.

KOHAVI, Ron. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, v. 2, 1995. p. 1137–1143.

MITCHELL, Tom M. *Machine Learning*. New York: McGraw-Hill, 1997.

NEWZOO. *Global Esports & Live Streaming Market Report 2022*. Free version. Amsterdam: Newzoo, 2022. Disponível em: `https://investgame.net/wp-content/uploads/2023/08/2022_Newzoo_Free_Global_Esports_Live_Streaming_Market_Report.pdf`. Acesso em: 18 maio 2026.

PEDREGOSA, Fabian et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

REJTHAR, Jan; KOTRBA, Vojtěch. Confirmation of the validity of the HLTV ranking. *ICGA Journal*, v. 45, n. 1, p. 2–15, 2023. DOI: 10.3233/ICG-230229.

SHENKMAN, Evgeniya; MOLODCHIK, Mariia; PARSHAKOV, Petr. Online Versus Offline eSports Tournaments: Prize Structure and Individual Performance. *Journal of Sports Economics*, v. 26, n. 1, p. 115–132, 2025. DOI: 10.1177/15270025241284056.

ŠVEC, Ondřej. *Predicting Counter-Strike Game Outcomes with Machine Learning*. Projeto de graduação, Czech Technical University in Prague, Faculty of Electrical Engineering, 2022.

XENOPOULOS, Peter; DORAISWAMY, Harish; SILVA, Cláudio T. Valuing Player Actions in Counter-Strike: Global Offensive. In: *2020 IEEE International Conference on Big Data (Big Data)*. Atlanta: IEEE, 2020. p. 1283–1292. DOI: 10.1109/BigData50022.2020.9378154.

XIA, Xinyao; SALINAS, Abel; MORSTATTER, Fred. Precision Under Fire: Analysis and Predictive Modeling in Counter-Strike: Global Offensive. In: *2025 IEEE Conference on Games (CoG)*. Lisbon: IEEE, 2025. p. 1–8. DOI: 10.1109/CoG64752.2025.11114164.

APÊNDICE A — ATRIBUTOS PRÉ-JOGO CONSIDERADOS

A Tabela 12 lista os 38 atributos pré-jogo considerados na bateria experimental, na ordem em que aparecem na base de dados.

Tabela 12 – Atributos pré-jogo considerados na bateria experimental

Identificador na base	Descrição
`diff_recent_win_rate`	**Taxa recente de vitórias**
`diff_maps_played_mean_5`	Mapas jogados
`diff_rounds_played_mean_5`	*Rounds* jogados
`diff_total_kills_mean_5`	Total de abates
`diff_rating_mean_5`	***rating* médio dos jogadores**
`diff_adr_mean_5`	Dano médio por *round* (ADR)
`diff_impact_mean_5`	***impact* médio**
`diff_kast_mean_5`	**KAST médio**
`diff_kd_mean_5`	**Razão entre abates e mortes**
`diff_kills_per_round_mean_5`	Abates por *round*
`diff_deaths_per_round_mean_5`	**Mortes por *round***
`diff_assists_per_round_mean_5`	Assistências por *round*
`diff_headshot_pct_mean_5`	Percentual de *headshots*
`diff_saved_by_teammate_per_round_mean_5`	Vezes salvo por companheiro por *round*
`diff_utility_damage_per_round_mean_5`	Dano de utilitário por *round*
`diff_flash_assists_per_round_mean_5`	Assistências de *flash* por *round*
`diff_utility_kills_per_100_rounds_mean_5`	Abates com utilitário por 100 *rounds*
`diff_time_opponent_flashed_per_round_mean_5`	Tempo de oponente cegado por *round*
`diff_opening_kills_per_round_mean_5`	**Abates de abertura por *round***
`diff_opening_deaths_per_round_mean_5`	Mortes de abertura por *round*
`diff_opening_success_mean_5`	**Sucesso em aberturas**
`diff_win_after_opening_kill_pct_mean_5`	*Rounds* vencidos após abate de abertura (%)
`diff_trade_kills_per_round_mean_5`	Abates de troca (*trade*) por *round*
`diff_saved_teammate_per_round_mean_5`	Companheiros salvos por *round*
`diff_last_alive_pct_mean_5`	**Percentual de último jogador vivo**
`diff_one_on_one_win_pct_mean_5`	Vitórias em duelos individuais (%)
`diff_time_alive_per_round_mean_5`	**Tempo vivo por *round***
`diff_rounds_with_a_kill_mean_5`	*Rounds* com ao menos um abate
`diff_kills_per_round_win_mean_5`	Abates por *round* vencido
`diff_damage_per_round_win_mean_5`	Dano por *round* vencido
`diff_pistol_round_rating_mean_5`	*rating* em *rounds* de pistola
`diff_players_count`	Jogadores disponíveis no *snapshot*
`diff_missing_players`	Jogadores ausentes no *snapshot*
`context_is_lan`	Indicador de partida LAN
`context_is_bo1`	Indicador de série melhor-de-1
`context_is_bo3`	Indicador de série melhor-de-3
`context_is_bo5`	Indicador de série melhor-de-5
`diff_elo_pre_match`	**ELO pré-jogo**

Fonte: elaboração própria. *Nota:* `diff_` indica diferença entre a equipe analisada e a adversária; `mean_5`, média dos até cinco jogadores do *snapshot* sazonal; `context_`, condição binária da partida. Os 11 atributos em negrito compõem o conjunto *univariate top 11* do modelo final (Figura 5); contexto e elenco participaram como candidatos, mas não foram selecionados.