Instituto de Biociências, Letras e Ciências Exatas, Unesp
Departamento de Ciência da Computação e Estatística

ANÁLISE DE DIFERENTES ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA RECOMENDAÇÃO DE RESULTADOS NA FRANQUIA COUNTER STRIKE

Carlos Eduardo Nogueiro Silva
Felipe Gomes da Silva
Felipe Matheus Possari
Renan Sinhorini Pimentel

Novembro
2024

Contents

1 Introdução

Durante anos, os e-sports, apesar de atuarem no âmbito eletrônico, não utilizaram amplamente as estatísticas e dados de jogos para embasar as decisões táticas e/ou não esportivas. Porém, recentemente, esses dados foram fortemente inseridos no cenário competitivo profissional, especialmente na franquia Counter-Strike, que é o nosso objeto de estudo.

O volume de dados gerado nas grandes competições promove, de forma quase intuitiva, a necessidade de estabelecermos modelos precisos que analisem o desempenho de grandes nomes do cenário. Assim, podemos interpretá-los em diferentes âmbitos, seja para avaliações técnicas e táticas ou para resultados esportivos, gerando ao usuário do modelo uma menor incerteza sobre resultados que transitam entre o acaso do esporte e a precisão das estatísticas.

1.1 Objetivos

O presente trabalho aborda a problemática de como diferentes algoritmos de aprendizado de máquina se comportam quando treinados com dados de Counter Strike, utilizando um sistema de rating. O objetivo principal é determinar a probabilidade de vitória de ambos os times, uma vez que essa informação pode ser útil para inúmeros casos, assim como em apostas esportivas.

2 Justificativa

A escolha do tema foi pensada para oferecer uma vantagem significativa em termos de estratégia, seja para os próprios times, analistas ou até mesmo para o público que acompanha o cenário competitivo e realiza apostas. A estatística desempenha um papel fundamental neste estudo, pois possibilita a modelagem da incerteza e a análise de tendências com base em dados concretos.

De acordo com o estudo de Ondřej Švec (2022), um modelo baseado em ELO seria a melhor escolha para a predição de resultados, podendo alcançar até 64% de acurácia. Faz-se necessário, então, unir dados reais e avaliações anteriores, como o ELO, em busca de uma recomendação precisa de resultados.

3 5W1H

What – O que vamos fazer?

Faremos um estudo da base de dados do Kaggle:[5], que contém mais de três mil partidas profissionais de Counter Strike. Utilizando estes dados escolheremos um modelo de classificação baseado em aprendizado de máquina para a análise dos dados. Uma vez feita essa escolha, usaremos esse modelo para gerar predições de resultados de partidas de Counter Strike.

Where – Para onde estamos olhando?

Utilizaremos dados de times do mais alto nível de competição do jogo, os 30 melhores times, ranqueados pelo site hltv.org, uma plataforma muito conhecida no meio competitivo de CSGO, dos maiores campeonatos dos anos em questão. O estudo terá ênfase nas estatísticas gerais do time, usando parcilamente dados individuais de cada jogador, uma vez que esses dados não refletem corretamente o desempenho da equipe como um todo.

When – Para quando estamos olhando?

A base escolhida possui dados de partidas jogadas em grandes campeonatos entre os anos 2016 e 2020, portanto o estudo busca prever os resultados das partidas dos anos posteriores a 2020.

Who – Para quem é este estudo?

O público alvo desse estudo são usuários de plataformas de aposta, em específico as que possuem a categoria Counter Strike, uma vez que com os resultados que buscamos, será possível prever ganhadores de futuras partidas. Dessa forma esperamos poder guiar os apostadores a maiores lucros. Mesmo não sendo o público alvo principal, também buscamos auxiliar jogadores de times profissionais e suas respectivas comissões técnicas, uma vez que não analisaremos apenas resultados e sim dados gerais dos times.

Why – Por que estamos fazendo este estudo?

O motivo da escolha do tópico se dá não só pela afinidade que possuímos com o jogo, mas principalmente da escassez de estudos estatísticos e probabilísticos aprofundados nessa área. Com o aumento da popularidade não somente do jogo mas também das plataformas de aposta, possuímos um grande incentivo e um público cada vez maior capaz de usufruir dos resultados e análises do nosso estudo.

How - Como vamos fazer?

Primeiramente faremos o pré-processamento dos dados para filtrar apenas os dados que realmente importam para a análise.

Após isso, utilizaremos um sistema Elo para cada um dos times com base em fórmulas estabelecidas por Arpard Elo, inicialmente criadas para classificar jogadores de xadrez. Realizando alguns ajustes nas aplicações das fórmulas, usaremos um modelo de classificação com aprendizado de máquina para realizar as predições por meio do treinamento do modelo com os dados que possuímos.

4 Fonte de Dados

Inicialmente, tentou-se coletar dados através de **web scraping** de sites especializados, como o **HLTV**, que fornece estatísticas detalhadas de partidas profissionais. No entanto, devido a restrições impostas pelo site (banimento por IP) por muitas requisições ao servidor, foi necessário alterar a estratégia de coleta. Optou-se, então, por utilizar o "CS* Professional Matches Dataset" disponível no Kaggle disponível em: [5], que oferece um conjunto abrangente de dados das partidas entre os principais times profissionais.

4.1 Descrição do Dataset

- **Período Coberto:** 2016 a 2020.
- **Times:** Partidas envolvendo os times presentes no top 20 do ranking global durante o período.
- **Estatísticas Individuais:** Dados como **rating**, **impacto**, **taxa de kills por morte (KDR)**, **dano médio por round (DMR)**, **kills por round (KPR)**, **assistências por round (APR)**, entre outros.
- **Informações dos Times:** Inclui o **ranking mundial**, **porcentagem de vitórias em confrontos diretos (head-to-head)**, **Elo**, entre outros.

5 Pré-processamento dos dados

Foram removidas do dataset todas as informações que só estariam disponíveis após o término das partidas, como os **pontos finais de cada time** e o **resultado da partida**. Isso evita a introdução de viés e garante que o modelo utilize apenas informações que estariam disponíveis antes do início de uma partida real.

5.1 Cálculo do Elo dos Times

O sistema Elo, originalmente desenvolvido por Arpad Elo para ranquear jogadores de xadrez, foi adaptado para diversas competições e esportes ao longo dos anos, incluindo o cenário competitivo de **Counter-Strike: Global Offensive (CS)**. Este sistema foi incorporado ao estudo como uma ferramenta fundamental para representar a habilidade relativa das equipes em cada partida.

5.2 Funcionamento do Elo

Inicialmente, todos os jogadores começam com um rating base de **1500**. A cada partida, o rating dos jogadores é ajustado com base no resultado da partida e na diferença de habilidade entre os oponentes, calculada pelo próprio sistema. O cálculo do novo Elo é feito através das seguintes fórmulas:

5.2.1 Cálculo da Probabilidade de Vitória de A (EA):

$$EA = \frac{1}{1 + 10^{(RB-RA)/400}}$$

Onde:

- **RA**: Elo do time A antes da partida.
- **RB**: Elo do time B antes da partida.

Esta fórmula calcula a probabilidade de vitória de uma equipe com base nas pontuações iniciais.

5.2.2 Atualização do Elo:

$$RA' = RA + K \cdot (SA - EA)$$

Onde:

- **RA'**: Novo Elo do time A após a partida.
- **SA**: Resultado da partida para o time A (1 para vitória, 0 para derrota, 0.5 para empate).
- **K**: Fator de ajuste que controla o impacto do resultado na pontuação.

O novo Elo é determinado ajustando RA (o Elo inicial) com base no resultado esperado (EA) e no resultado real (SA), multiplicado pelo fator de ajuste K.

Neste estudo, o Elo foi utilizado como uma métrica para representar a força relativa das equipes antes de cada partida. Para isso, o Elo de uma equipe foi calculado como a média dos ratings dos jogadores titulares.

Após cada partida, a diferença do novo Elo foi somada ao rating de cada jogador, permitindo que o sistema acompanhasse a evolução individual e coletiva das equipes. Essa abordagem permitiu capturar o desempenho histórico e a progressão das equipes, fornecendo uma base sólida para uma análise comparativa.

5.3 Cálculos das features

5.3.1 Médias das Estatísticas Individuais

Para cada estatística individual dos jogadores, como **impacto**, **KDR**, **DMR**, entre outras, foi calculada a média dos cinco jogadores titulares de cada time. Isso resultou em novas features representando o desempenho médio do time em cada aspecto, simplificando a análise e reduzindo a dimensionalidade dos dados.

5.3.2 Diferenças Entre os Times

Para quantificar a **diferença de desempenho** entre os times, foi calculada a diferença entre as médias das estatísticas dos times em cada partida. Além disso, foram calculadas as diferenças no **ranking mundial** e no **Elo**. Essas diferenças servem como indicadores de vantagem ou desvantagem relativa de um time sobre o outro.

5.3.3 Seleção de Features

Foram removidas colunas irrelevantes para a predição, como **nomes dos times** e **datas das partidas**. O foco foi mantido em variáveis numéricas e categóricas que poderiam influenciar diretamente o resultado.

6 Seleção do Modelo

6.1 Algoritmos Testados

Foram testados alguns algoritmos de classificação, sendo eles:

1. **Naive Bayes:** Baseado no Teorema de Bayes, assume independência entre as features.
2. **Regressão Logística:** Modelo estatístico para classificação binária.
3. **K-Nearest Neighbors (KNN):** Classifica uma observação com base nos K vizinhos mais próximos.
4. **Support Vector Machine (SVM):** Encontra um hiperplano que melhor separa as classes.
5. **Random Forest:** Ensemble de árvores de decisão que melhora a precisão e reduz overfitting.

6.2 Validação Cruzada

Utilizou-se o método de **K-Folds Cross-Validation**, com **K=10**, para avaliar o desempenho dos modelos. A média das acurácias obtidas foi utilizada como critério de comparação.

6.3 Resultado

O **Random Forest** obteve a melhor performance no teste de validação, mostrando-se mais eficaz em capturar as complexidades dos dados em comparação com os outros algoritmos, portanto, foi o escolhido.

7 Análise de Dados

7.1 Otimização de Hiperparâmetros

Após identificar o **Random Forest** como o modelo mais promissor, realizou-se uma **otimização de hiperparâmetros** utilizando o **Random Search**. Esse método seleciona aleatoriamente combinações de hiperparâmetros a serem testadas, sendo computacionalmente mais eficiente que o Grid Search.

Hiperparâmetros Otimizados

- **n_estimators:** Número de árvores na floresta.
- **max_features:** Número de features consideradas em cada divisão.
- **max_depth:** Profundidade máxima das árvores.
- **min_samples_split:** Mínimo de amostras para dividir um nó.
- **min_samples_leaf:** Mínimo de amostras em uma folha.
- **bootstrap:** Uso ou não de amostragem com reposição.

Configuração do Random Search

- **Número de Iterações (n_iter):** 100 combinações de hiperparâmetros.
- **Validação Cruzada:** Utilizou-se o `RepeatedStratifiedKFold` com `n_splits=10` e `n_repeats=5` para aumentar a robustez.
- **Métrica de Avaliação:** Acurácia.

7.2 Implementação e Treinamento

O modelo foi treinado utilizando o **Scikit-Learn**, segundo as melhores práticas de pré-processamento e validação. Foi utilizado o `RandomizedSearchCV` para automatizar o processo de escolha de hiperparâmetros ótimos.

8 Análise dos Resultados

Após a otimização, o modelo **Random Forest** alcançou uma **acurácia de 69%** no conjunto de teste. A acurácia obtida indica que o modelo é capaz de prever corretamente o resultado das partidas em 69% dos casos. Considerando a natureza competitiva e imprevisível dos esportes eletrônicos, esse resultado é significativo.

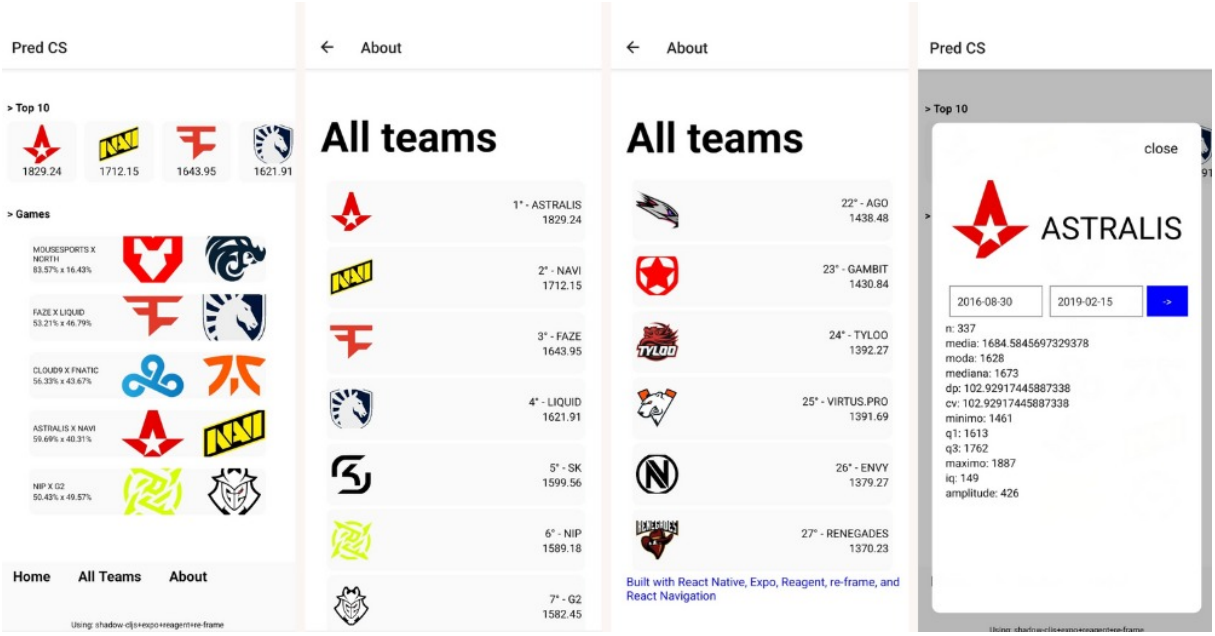
Fatores que podem ter influenciado o desempenho:

- **Qualidade dos Dados:** A precisão das estatísticas e a cobertura temporal podem afetar a capacidade preditiva.
- **Complexidade do Jogo:** CS é um jogo com variáveis táticas e estratégicas difíceis de quantificar.
- **Limitações do Modelo:** Mesmo modelos avançados têm dificuldade em capturar todas as nuances que levam à vitória ou derrota em uma partida.

9 Interface gráfica

Para a interface gráfica desenvolvemos um aplicativo mobile para visualização dos resultados do projeto, nele é possível consultar estatísticas dos times, visualizar probabilidades de vitória e também consultar o ranking de melhores times ranqueados por elo, como pode ser visto nas imagens abaixo:

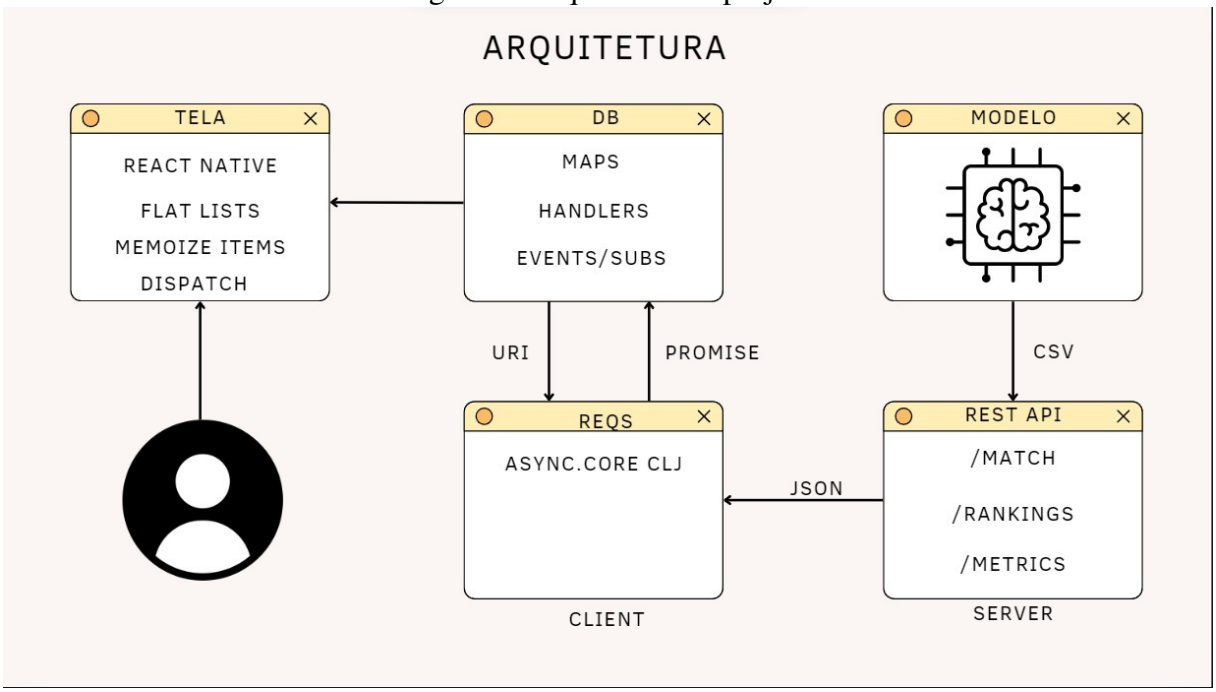
Figura 1: Imagem da UI do aplicativo



Fonte: Autoral

O diagrama abaixo representa um fluxo de dados em um sistema que utiliza uma interface móvel (app) conectada a um backend via API REST, com funcionalidades de manipulação de dados, estado e lógica.

Figura 2: Arquitetura do projeto



Fonte: Autoral

9.1 Destaques

Esta seção não contempla uma análise detalhadaa acerca do funcionamento das tecnologias que utlizou-se no projeto, entretanto, visa identificar as escolhas e o funcionamento geral do

projeto.

9.1.1 Tela (Interface do Usuário):

- **Flat Lists:** Uma estrutura usada para exibir listas de dados de forma eficiente (por exemplo, uma lista de partidas esportivas).
- **Memoize Items:** Técnica para otimizar o desempenho, reutilizando itens já calculados ou renderizados.
- **Dispatch:** O envio de eventos (ações) que pedem mudanças no estado global da aplicação.

⇒ *A Tela recebe os dados do banco de estado (DB) e envia eventos (ações) para modificar o sistema em paralelo a propria renderização.*

9.1.2 DB (Banco de Estado):

- **Handlers:** Funções que processam eventos enviados pela tela (ex.: atualizar um placar).
- **Events/Subs:** Sistema de eventos (input) e assinaturas (output).
  - *Eventos:* Comandos que modificam o estado.
  - *Subscrições:* Consultas para acessar dados do estado.

⇒ *O DB é o coração da aplicação no lado do cliente, conectando a interface com o restante do sistema.*

9.1.3 Requisições:

- **async.core:** Ferramentas em ClojureScript para lidar com chamadas assíncronas, como comunicação com a API.
- **Promise:** Uma estrutura para lidar com respostas de chamadas assíncronas (aguardando dados do servidor).

⇒ *As REQS enviam e recebem dados da API REST, conectando o app com o servidor.*

9.1.4 Modelo:

- Representa a lógica mais complexa do sistema.
- **CSV:** Arquivos usados para alimentar o modelo com dados brutos. Ademais, foram usasdos para transimitir os dados processados pelo modelo.

⇒ *Ele alimenta o backend (API REST) com dados processados.*

9.1.5 REST API (Servidor):

- Disponibiliza rotas como /match, /rankings e /metrics para fornecer ou receber informações.
- O backend processa dados e envia respostas em JSON para o cliente.

⇒ *É o intermediário entre o modelo (dados e lógica) e o aplicativo no cliente.*

9.2 Fluxo de Dados

1. O usuário interage com a **Tela** (ex.: visualiza uma lista ou clica em um botão).
2. Um evento é ***dispatched*** para o **DB**, que processa a lógica necessária.
3. Se for necessário buscar dados externos, uma **requisição** é feita para a **REST API**.
4. A **API** consulta o **Modelo** e retorna uma resposta.
5. A resposta é processada no **DB**, e a **Tela** é atualizada com os novos dados.

10 Conclusão

O projeto foi concluído com êxito. Utilizou-se o modelo **Random Forest** para realizar a análise dos dados e observou-se uma satisfatória acurácia de 69%. Dessa forma, o objetivo previamente estabelecido foi alcançado, uma vez que, com esta margem de acerto, a gama de testes, a aplicação e o servidor funcionais, pode-se garantir uma clara vantagem ao usuário em relação a predição de vencedores de partidas de Counter Strike.

References

[1] CHALMERS. Predicting the outcome of CS games using machine learning. 2018. Disponível em: https://publications.lib.chalmers.se/records/fulltext/256129/256129.pdf. Acesso em: [01 out. 2024].

[2] LUND UNIVERSITY. Predicting Counter-Strike Matches. 2024. Disponível em: https://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=9145457. Acesso em: [02 out. 2024].

[3] HLTV.org: your source for esports information. Disponível em: https://www.hltv.org. Acesso em: [29 set. 2024].

[4] SVEC, O. Predicting Counter-Strike Game Outcomes with Machine Learning. Disponível em: https://dspace.cvut.cz/bitstream/handle/10467/99181/F3-BP-2022-Svec-Ondrej-predicting_csgo_outcomes_with_machine_learning.pdf. Acesso em: [02 out. 2024].

[5] https://www.kaggle.com/datasets/gabrieltardochi/counter-strike-global-offensive-matches. Acesso em: [20 nov.2024].