

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/321899230>

# Aplicação de aprendizado de máquina para otimização da escalação de time no jogo Cartola FC

Article · December 2017

CITATIONS  
0

READS  
234

3 authors, including:



Gabriel Viscondi  
University of São Paulo  
1 PUBLICATION 0 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



PCS5031 [View project](#)

# Aplicação de aprendizado de máquina para otimização da escalação de time no jogo Cartola FC<sup>1</sup>

VISCONDI, G. F.<sup>\*1</sup>, JUSTO, D.<sup>\*2</sup>, GARCÍA, N. M.<sup>\*3</sup>

<sup>\*</sup> Escola Politécnica da Universidade de São Paulo (POLI-USP) – Departamento de Computação e Sistemas Digitais

<sup>1</sup>gabrielviscondi@usp.br, <sup>2</sup>diogenes.justo@gmail.com, <sup>3</sup>nelsonmurciagarcia90@gmail.com

**Resumo** - A utilização de técnicas de aprendizado de máquina tem se expandido largamente, contribuindo consistentemente para a evolução dos modelos de predição em diversos setores de aplicação. Este artigo propõe a aplicação de modelos de aprendizado de máquina para predição da escalação do time capaz de atingir a maior pontuação no *fantasy game* Cartola FC, utilizando-se a base de dados histórica disponibilizada pelo próprio aplicativo. Utilizou-se os algoritmos *extreme gradient boosting* (XG), *random forest* (RF) e *support vector machine* (SVM) como métodos de predição, baseando-se nos dados de desempenho dos jogadores nas rodadas passadas. Ainda, utilizando-se uma técnica de aprendizado não supervisionado, *k-means*, foi proposta uma divisão dos jogadores em grupos de características semelhantes para aprimorar a acurácia do modelo. Os resultados preliminares demonstram que a utilização de dados fornecidos diretamente através da API não atingem resultados satisfatórios por meio da utilização dos algoritmos propostos, com performance ligeiramente superior do algoritmo RF. Entretanto, a clusterização proposta pelo algoritmo *k-means* conseguiu elevar razoavelmente a pontuação da equipe sugerida pelo modelo. Apesar da imprevisibilidade do comportamento dos atletas e dos resultados das partidas, o modelo reduz o erro de alocação para o jogador, encontrando resultados razoáveis em comparação com a média da população.

**Palavras-chave** – *aprendizado de máquina, random forest, extreme gradient boosting, clusterização, k-means*

## 1. INTRODUÇÃO

O uso da tecnologia tem sido extremamente efetivo no auxílio à tomada de decisão nos mais variados setores da economia, levando, dentre outros aspectos, ao aumento da produtividade, à redução de custos operacionais, ao aumento na acurácia de previsões e à maior agilidade nas decisões seja no poder público, no setor privado, na academia ou nas organizações do terceiro setor [1].

Essa nova perspectiva tecnológica vem acompanhada de uma distinta dimensão de coleta e interpretação de dados, visto o crescimento acelerado de novos ambientes e fontes de aquisição: redes sociais, sensores, e-mails, transações bancárias, jogos eletrônicos, dentre outros, que acabam trazendo novas perspectivas

de abordagem dos problemas [2]. Essa intensa concentração de dados traz consigo variabilidade, velocidade e volume [3], fazendo com que, neste âmbito, tecnologias como *crowd sourcing*, *machine learning* e *data mining*, por exemplo, surjam para minimizar riscos de análises e/ou auxiliar a extração de informações em *big data*.

Em jogos eletrônicos, esta tendência não é diferente. Além de apoiar o desenvolvimento destes aplicativos por meio, por exemplo, do aprimoramento da inteligência artificial dos seus mais variados elementos [4,5,6], os algoritmos de aprendizado de máquina têm desempenhado papel importante na compreensão das melhores estratégias de jogo, aprimorando a experiência do jogador e os resultados obtidos em cada partida [7,8,9].

Tentando prever o vencedor de uma partida no *starcraft*, um jogo de estratégia em tempo real, autores propõem uma abordagem baseada em algoritmos de aprendizado de máquina baseada em dados de jogadores reais e computadores [10]. Em outro trabalho, autores combinam algoritmos de mineração de dados e aprendizado de máquina para identificar padrões em jogos 2D e treinar jogadores virtuais para a vitória [11].

Este artigo apresenta uma metodologia de aplicação e comparação de dois algoritmos de aprendizado de máquina, na sugestão de escalação do melhor time para a próxima rodada no *fantasy game* Cartola FC. Sendo assim, após esta introdução, o artigo apresenta o funcionamento do jogo em seu segundo capítulo, uma breve descrição dos algoritmos utilizados no capítulo 3, os objetivos no capítulo 4 e, no capítulo 5, toda a metodologia desenvolvida. Em seguida, o capítulo 6 discorre sobre os resultados obtidos pelo estudo e, ao fim, as conclusões são tecidas no último capítulo.

## 2. APRESENTAÇÃO DO CARTOLA FC

O Cartola FC<sup>2</sup> é um jogo virtual vinculado à emissora de televisão Rede Globo, e que simula para seus jogadores – denominados internamente de cartoleiros – a possibilidade de escalação de

<sup>1</sup> Doi:

<sup>2</sup> O jogo Cartola FC pode ser acessado em: <http://globoesporte.globo.com/cartola-fc/>

uma equipe de futebol a cada rodada da série A do Campeonato Brasileiro.

Sendo assim, o jogador cadastrado cria a sua própria equipe no início do Campeonato Brasileiro, escolhendo seu nome e vinculando-a à sua conta no site.

Para a primeira rodada, cada jogador começa com 100 cartoletas (unidade monetária do jogo), as quais são utilizadas para a aquisição de 11 jogadores e um técnico de futebol. Vale ressaltar que, apesar de ser um jogo fantasia, todos os atletas e técnicos passíveis de escalação são os mesmos que participam do Campeonato Brasileiro de Futebol.

Dando continuidade ao jogo, cada cartoleiro deve executar as seguintes tarefas antes de cada rodada do Campeonato Brasileiro:

- 1) Escolher a formação tática de sua equipe (3-5-2, 4-4-2, ...);
- 2) De acordo com a formação, indicar os atletas para ocupar cada posição de acordo com o orçamento disponível em sua carteira na rodada;

Uma vez definidas todas as equipes, os jogos de futebol reais do Campeonato Brasileiro ocorrem e, para cada jogador e técnico diversos indicadores de performance são contabilizados e armazenados em uma base de dados. Esses indicadores, denominados *scouts*, podem ser positivos ou negativos, definindo por meio de um sistema de pontos a performance de cada atleta na rodada. A pontuação no final da rodada pode ser positiva, pelo acúmulo de *scouts* como gols e assistências, ou negativa se o jogador fizer inúmeras faltas, por exemplo. Os *scouts* e suas respectivas pontuações são:

- 1) **Positivos (+):** roubada de bola (1,7); gol (8); assistência (5); jogo sem sofrer gol (5); falta sofrida (0,5); finalização para fora (0,7); finalização defendida (1); finalização na trave (3,5); defesa difícil (3); defesa de pênalti (7).
- 2) **Negativos (-):** gol contra (6); cartão vermelho (5); cartão amarelo (2); gol sofrido (5); pênalti perdido (3,5); falta cometida (0,5); impedimento (0,5); passe errado (0,3).

A partir da pontuação na rodada anterior, o jogador pode ficar mais caro ou mais barato para ser escalado na rodada subsequente. Ainda, o somatório da pontuação dos atletas escalados por cada jogar é que define a pontuação no jogo de cada equipe virtual a cada rodada.

Ao final das 38 rodadas do Campeonato Brasileiro, é vitoriosa a equipe que tiver o maior número de pontos segundo os critérios acima descritos. Na edição de 2017, o jogo contou com mais de 4 milhões de times escalados durante suas rodadas iniciais e o jogador vencedor totalizou 2.562,41 pontos ao final de todas as rodadas (média de 67,4 pontos por rodada).

### 3. APRENDIZADO DE MÁQUINA PARA CLUSTERIZAÇÃO E PREVISÃO

Atualmente, o volume de dados adquiridos e a agilidade com que as decisões precisam ser tomadas exigem soluções que validem estatisticamente o conhecimento gerado a partir desta coleta. Em diversas soluções de ciência dos dados, técnicas de mineração de dados e aprendizado de máquina são utilizadas previamente para organizar e categorizar previamente toda a heterogeneidade de dados coletada [12].

Neste âmbito, o algoritmo de clusterização não supervisionada denominado k-means é largamente utilizado devido sua simplicidade de aplicação. Este algoritmo baseia-se na determinação de um conjunto de centroides (k) em uma base contendo “n” pontos de dados em um espaço dimensional definido. Desta forma, o objetivo do algoritmo é minimizar a distância quadrática média entre cada ponto de dados e o centroide mais próximo [13]. Dentre suas aplicações, estes algoritmos podem agrupar imagens, vídeos ou documentos de texto, ajudando em diversos problemas científicos como detecção de padrões em desenvolvimento de patogênicos, comportamento de veículos no tráfego e análise de dados compra no varejo.

Outro campo de aplicação de técnicas de ciência de dados se dá na previsibilidade e análise de tendências futuras. Estes algoritmos são capazes de analisar padrões e gerar hipóteses a partir de série de dados históricas e, a partir do desenvolvimento de uma série de correlações, prever com acurácia considerável tendências futuras de comportamento [14].

Para a elaboração deste artigo, selecionou-se três técnicas já consolidadas academicamente no desenvolvimento de previsões: *random forest* (RF), *extreme gradient boosting* (XG) e *support vector machines* (SVM).

Os algoritmos de RF fundamentam-se na combinação de árvores de predição, onde cada árvore depende dos valores de um vetor selecionado aleatoriamente a partir do universo de dados em que está sendo trabalhado e com a mesma distribuição de todas as árvores da “floresta”. O erro destes modelos depende da intensidade de correlação individual entre cada árvore da floresta considerada, convergindo para um limite assim que o número de árvores se expande [15].

Por sua vez, EGB é uma derivação mais eficiente e escalável do *framework* de *gradient boosting* baseado em árvores de decisão proposto por Friedman em 2001 [16]. Nestes algoritmos, o procedimento de aprendizado adapta constantemente novos modelos para estimar com maior precisão seus resultados. A principal ideia por trás desse algoritmo é construir novos

*learners* de base par serem correlacionados ao máximo com o gradiente negativo da função de perda, associado a todo o conjunto. Estas funções de perda aplicadas podem ser arbitrárias, mas para dar uma melhor intuição, se a função de erro for a perda de erro quadrático clássica, o procedimento de aprendizagem resultaria em um erro análogo [17].

As SVM constituem um conjunto de métodos de aprendizado supervisionado utilizados tanto para classificação como para regressão, pertencendo a uma família de classificadores lineares generalizados. O mapeamento do SVM cria um vetor em um espaço dimensional superior onde um hiperplano máximo de separação é construído. Assim, outros dois hiperplanos são estabelecidos em cada lado do primeiro, separando os dados e fazendo com o que o central maximize a distância entre os dois hiperplanos paralelos. Logo, o algoritmo funciona a partir da suposição que quanto maior a margem ou a distância entre esses hiperplanos, maior será o erro de generalização do classificador [18].

Este método vem sido amplamente estudado pela comunidade científica devido aos bons resultados de predição e seu amplo espectro de aplicação. Trabalhos apontam seu sucesso de aproveitamento em diversos problemas como reconhecimento de tom, reconhecimento de escrita manual, categorização de texto, detecção de objetos, dentre outros [18].

#### 4. OBJETIVOS

Este artigo possui como objetivo principal apresentar algoritmos para prever, por meio de análise de dados de pontuação dos jogadores nas rodadas passadas, a possível escalação ótima (que conseguirá o maior número de pontos) para a rodada vigente do jogo Cartola FC.

Vale ressaltar que o objetivo por detrás da aplicação destes algoritmos neste trabalho, não é prever corretamente qual será a pontuação devido a escalação de um determinado conjunto de jogadores. A ideia central do trabalho está em reduzir o risco de escalação dos jogadores, sugerindo um grupo de atletas de maneira que a pontuação seja maximizada pelo uso do algoritmo.

A metodologia, descrita em detalhes na seção 5 deste artigo, fundamenta-se na aquisição dos dados do jogo Cartola FC por meio de API fornecida oficialmente pelos desenvolvedores. Em seguida, três algoritmos serão propostos para investigar a possibilidade de predição da escalação ótima do time: RF, EGB e SVM. Em ambos modelos de predição, os dados serão

previamente clusterizados pelo algoritmo *k-means*.

Sendo assim, em complemento ao objetivo principal de definir metodologia de previsão, os autores propõem também analisar, comparativamente, a acurácia e tempo de processamento dos resultados de previsão entre os algoritmos RF, EGB e SVM.

### 5. METODOLOGIA

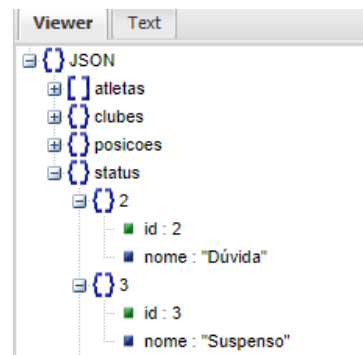
#### 5.1. Aquisição e Armazenamento dos Dados

Como descrito na Seção 2 deste artigo, o jogo Cartola FC conta com a participação de 20 equipes da série A do campeonato brasileiro, as quais disputam duas partidas entre si – uma em seu estádio e outra como visitante.

Sendo assim, anualmente, o jogo possui duração de 38 rodadas compostas por 10 partidas cada. Atualizadas para cada uma destas rodadas, os desenvolvedores da aplicação disponibilizam a API<sup>3</sup> contendo todos os dados de caracterização da performance dos jogadores. A API vem em formato JSON, disponibilizando dados estruturados em quatro categorias:

- 1) **Atletas:** informações de cada um dos jogadores de futebol participando do campeonato, incluindo valor de contratação, número de jogos, foto, média de pontos, etc...
- 2) **Clubes:** informações de cada um dos 20 clubes da série A do Campeonato Brasileiro, como nome, posição, foto do escudo, etc...
- 3) **Posições:** descrição das diferentes posições ocupadas pelos jogadores de futebol: goleiro, lateral, atacante, dentre outros.
- 4) **Status:** definições possíveis de status atual dos jogadores para a rodada como suspenso, contundido, provável escalação, etc...

A **Figura 1** ilustra a maneira com que os dados são organizados, assim como a distribuição de sua estrutura:



**Figura 1** - Estrutura dos dados fornecidos pela API do jogo CartolaFC. Visualização em JasonViewer<sup>4</sup>

<sup>3</sup> API disponível gratuitamente em: <https://api.cartola.fc/globo.com/atletas/mercado>

<sup>4</sup> Visualização disponível em: <http://jsonviewer.stack.hu>

Utilizando-se da API, os dados são coletados a cada rodada em formato \*.csv e, por meio de um algoritmo escrito na linguagem R – *CargaeTratamentoDados.R* – os dados são atualizados em um arquivo único que contém informações de todo o decorrer do campeonato.

## 5.2 Algoritmos de Previsão

Os algoritmos foram implementados no software livre R-Studios<sup>5</sup>. Primeiramente, os dados foram pré-processados para eliminar registros de jogadores que não jogaram na partida. Inicialmente o arquivo continha 30.763 registros (quantidade de jogadores convocados para cada jogo), após o pré-processamento restaram 10.519 registros (eliminação de cerca de 66% dos dados).

Após o processamento das informações, utilizou-se um algoritmo *K-means* para agrupar os conjuntos de dados com valores semelhantes. Neste caso, o algoritmo agrupou por similaridade jogadores de futebol que tem o mesmo comportamento nos jogos, pontuações associadas aos *scouts*, dentre outros aspectos. Estima-se que se agrupando os jogadores com comportamentos semelhantes, pode-se obter melhores resultados na hora de prever a pontuação dos jogadores.

Em particular neste algoritmo, utilizou-se uma variante chamada *K-means ++*, descrita por [19]. Esta nova variante utiliza o logaritmo *k* para obter um agrupamento ótimo, além de executar o algoritmo *K-means* por várias vezes e manter apenas o de melhor resultado. Depois de alguns testes obtiveram-se os melhores resultados com  $K=21, 22, 23$  para as diferentes formações de jogo, 7 em total (3-4-3, 3-5-2, 4-3-3, 4-4-2, 4-5-1, 5-3-2 e 5-4-1).

Após a fase de clusterização, empregou-se três algoritmos para obter as previsões do time com maior número de pontos.

O primeiro algoritmo utilizado foi o XG, e, para fazer as previsões, escolheu-se a função *xgboost* da biblioteca do mesmo nome. Os parâmetros e informações escolhidas para essa função foram os seguintes: dados de teste e dados de treinamento do modelo; taxa de aprendizagem configurada igual a 1 para obter um cálculo com rapidez da função.

Em seguida utilizou-se o segundo algoritmo de predição: RF. Para este algoritmo escolheu-se a função *randomForest* da biblioteca do mesmo nome escolhendo-se o valor de 500 para o parâmetro de profundidade máxima da árvore e limitando-a em profundidade não muito alta.

Por fim, repetiu-se o experimento utilizando-se o SVM, por meio da biblioteca *e1071*,

utilizaram-se os parâmetros padrões ao executá-lo.

## 6. RESULTADOS E DISCUSSÃO

O modelo de jogo utilizado no Cartola é responsável por fornecer o primeiro parâmetro de decisão: a formação tática. Avalia-se, portanto, cada resultado em comparação a cada formação com o intuito de escolher qual a formação é preferível. O segundo parâmetro será o algoritmo de aprendizado escolhido. Com base nestes dois parâmetros, pode-se buscar também encontrar a resposta para a questão previsibilidade. Isto é, se for possível tomar uma decisão de uso de algum algoritmo para obter a maximização dos resultados.

Buscou-se um cenário inicial para comparação com as seguintes características:

- Previsão das rodadas 29 a 34 (5 rodadas)
- Base de treino: rodadas 1 a 28
- Base de teste: rodadas 29 a 34

Ambos os algoritmos utilizaram da mesma base de treinamento, mesma base de teste e mesmas variáveis. Aplicaram-se os três algoritmos com resultados sobre a base de testes citada acima, sendo medidos os resultados por meio da pontuação real para a escalação de 11 jogadores e 1 técnico sugerida como ótima pelo algoritmo.

Inicialmente observou-se, através da análise descritiva, os resultados das médias de pontuações alcançadas após a aplicação de cada algoritmo para cada formação, com objetivo de encontrar o algoritmo que apresente a maior média. Na Tabela 1 pode-se observar os resultados:

Algoritmos	Formações							Média	Algoritmo
	343	352	433	442	451	532	541		
RF	62,02	60,58	59,43	64,08	61,58	58,33	61,92		61,13
SVM	55,56	56,79	53,39	56,69	60,19	51,88	57,44		55,99
XGBoost	50,25	49,12	45,90	50,20	49,95	47,47	52,65		49,37
Média	55,94	55,50	52,91	56,99	57,24	52,56	57,34		55,50

Tabela 1 - Comparativo de Médias de Resultados

Na Figura 2 pode-se ver que o algoritmo RF apresentou os melhores resultados a despeito da maior variabilidade, sendo seguido pelo algoritmo SVM e por último os resultados do algoritmo XG.

<sup>5</sup> Todos os algoritmos desenvolvidos e citados durante este artigo estão disponíveis em: <https://github.com/diogenesjusto/PCS5031>

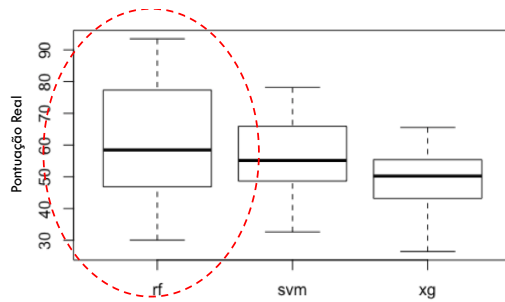


Figura 2 - Resultados por Algoritmo

Buscou-se também avaliar qual seria a formação ótima dado o emprego do algoritmo RF. Para isto, analisou-se o resultado da pontuação real caso fosse realizada a escalação sugerida como ótima pelo algoritmo RF, segmentando por formação. Na Figura 3 se pode observar que a variação entre as sete formações possíveis é pequena, porém com uma leve preferência para a formação 4-4-2.

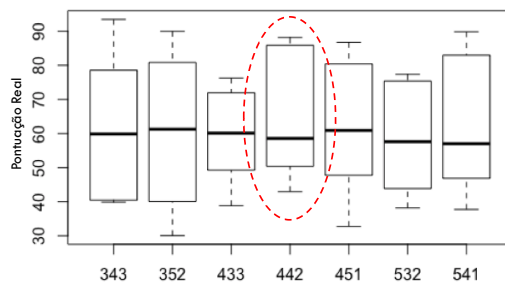


Figura 3 - Resultados por Formação

Procurou-se também entender o impacto da adição das variáveis nos modelos, com objetivo de compreender qual característica tem maior impacto na pontuação (performance) do jogador. Para isto, foram escolhidas as funções de análise de importância (relevância) de cada um dos algoritmos.

Para o algoritmo XG, encontra-se disponível a função `xgb.importance`, que lista para todas as variáveis utilizadas a relevância através da métrica *gain*, com valores possíveis de 0 a 1, que pode ser vista na Figura 4.

Para o algoritmo RF, a função `varImpPlot`, que gera o gráfico de importância para todas as variáveis utilizadas através da métrica *In Node Purity*, que pode ser visto na Figura 5.

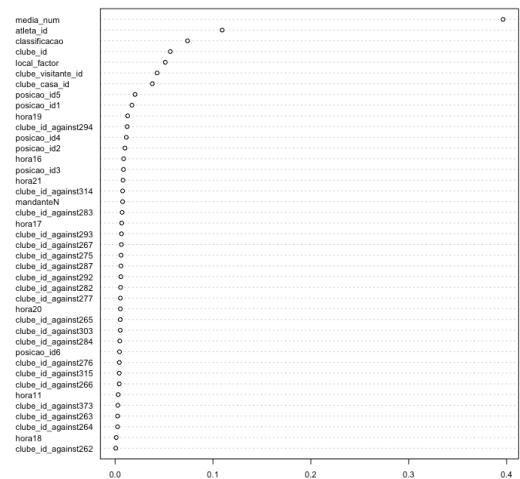


Figura 4 - Relevância das variáveis - XG

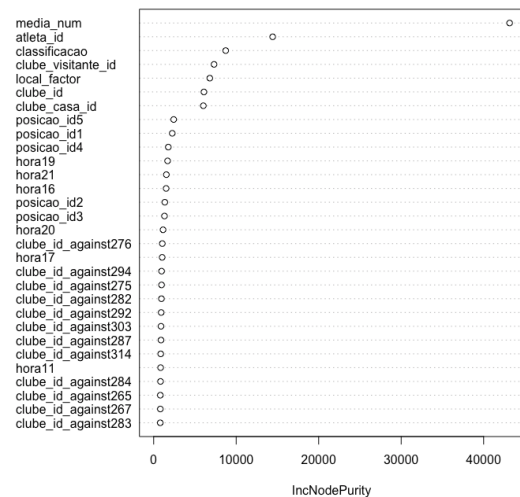


Figura 5 - Relevância das Variáveis - RF

Já o algoritmo SVM, no *package* `e1071`, não possui uma forma de analisar-se a importância das variáveis.

A partir desta análise é possível verificar que ambos algoritmos sinalizam as seguintes variáveis como de maior relevância: `media_num` (média de pontuação dos jogadores), `atleta_id` (identificação do jogador), `clube_visitante_id` (sinalização do clube que está jogando com seu mando de campo), `posicao_id` (posição na qual o jogador atua), `hora` (horário do jogo), `clube_id_against` (sinalização do clube que está jogando no campo adversário).

## 7. CONCLUSÕES E TRABALHOS FUTUROS

Os resultados apresentados pelos algoritmos de previsão – RF, XGB e SVM-, assim como a metodologia aplicada podem ser considerados satisfatórios para uma primeira iniciativa de mapeamento de tecnologias para desenvolver

modelos de predição associados ao jogo Cartola FC.

A despeito da imprevisibilidade de um jogo como o futebol, a modelagem empregada demonstrou resultado razoavelmente satisfatório o que se julga fazer sentido dada a característica de diluição de risco que é tentativa de previsão de resultados não de um jogador, mas de vários.

Como trabalhos futuros podem-se adicionar novos algoritmos de predição de resultados, como: *Árvore C4.5* (Weka J48), Redes Neurais de Convolução, dentre outros, buscando o aumento de performance do modelo e concluir qual algoritmo trabalha melhor nesta aplicação. Outra alternativa poderia ser compilar os dados dos jogadores em outras temporadas brasileiras, avaliando a variação da performance desses atletas em função do avanço da temporada.

Pode-se, também, cruzar os dados da API oficial do Cartola FC com outros dados, tais como

- Dados meteorológicos das partidas para avaliar desempenhos individuais em condições meteorológicas distintas;
- Notícias da web para obter informação sobre eventos externos dos jogos e que podem envolver informações sobre estados físicos e emocionais dos jogadores, assim como atrasos salariais e outros conflitos no clube, dentre outros;
- Dados de jogadores que não participam do campeonato brasileiro para enriquecer a base e delimitar comportamento semelhantes entre jogadores.

Por fim, acredita-se que estes resultados poderiam ser utilizados não só no jogo virtual Cartola FC, mas também para ajudar na tomada de decisões dos treinadores de times de futebol profissionais.

## REFERÊNCIAS

- [1] MCKINSEY & COMPANY. Big data: The next frontier for innovation, competition, and productivity. **McKinsey Global Institute**, n. June, p. 156, 2011;
- [2] ZHANG, Du. "Inconsistencies in Big data." 2013 IEEE 12<sup>th</sup> International Conference on Cognitive Informatics and Cognitive Computing, 2013, doi: 10.1109/icci-cc.2013.662222;
- [3] EDJLALI, R. (2011). Getting value from big data (webinar). Disponível em: <https://www.gartner.com/webinar/2056217>;
- [4] Galway, Leo, et al. "Machine Learning in Digital Games: A Survey." *Artificial Intelligence Review*, vol. 29, no. 2, 2008, pp. 123–161., doi:10.1007/s10462-009-9112-y;
- [5] Dobrovsky, A., Borghoff, U. and Hofmann, M. (2017). Applying and Augmenting Deep Reinforcement Learning in Serious Games through Interaction. *Periodica Polytechnica Electrical Engineering and Computer Science*, 61(2), p.198;
- [6] Nogueira-Collazo, Mariela, et al. "Competitive Algorithms for Coevolving Both Game Content and AI. A Case Study: Planet Wars." *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 8, no. 4, 2016, pp. 325–337., doi:10.1109/tciaig.2015.2499281;
- [7] Ontanón, S., Synnaeve, G., Uriarte, A., Richoux, F., Churchill, D., Preuss, M.: A survey of real-time strategy game AI research and competition in starcraft. *IEEE Trans. Comput. Intell. AI Games* 5(4), 293–311 (2013);
- [8] Synnaeve, G., Bessiere, P.: A Bayesian model for opening prediction in RTS games with application to StarCraft. In: *IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 281–288. IEEE (2011);
- [9] Semenov, Aleksandr, et al. "Performance of Machine Learning Algorithms in Predicting Game Outcome from Drafts in Dota 2." *Communications in Computer and Information Science Analysis of Images, Social Networks and Texts*, 2017, pp. 26–37., doi:10.1007/978-3-319-52920-2\_3;
- [10] Sanchez-Ruiz, Antonio A., and Maximiliano Miranda. "A Machine Learning Approach to Predict the Winner in StarCraft Based on Influence Maps." *Entertainment Computing*, vol. 19, 2017, pp. 29–41., doi:10.1016/j.entcom.2016.11.005;
- [11] Jacob, Lidson, et al. "Oh Gosh!! Why Is This Game so Hard? Identifying Cycle Patterns in 2D Platform Games Using Provenance Data." *Entertainment Computing*, vol. 19, 2017, pp. 65–81., doi:10.1016/j.entcom.2016.12.002;
- [12] Somvanshi, Madan, and Pranjali Chavan. "A Review of Machine Learning Techniques Using Decision Tree and Support Vector Machine." 2016 International Conference on Computing Communication Control and Automation (ICCUBEA), 2016, doi:10.1109/iccubea.2016.7860040. doi : 10.1109/TPAMI.2002.1017616;
- [13] Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y. An efficient k-means clustering algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Volume: 24, Issue: 7, Jul 2002);
- [14] Praveena, M., and V. Jaiganesh. "A Literature Review on Supervised Machine Learning Algorithms and Boosting Process." *International Journal of Computer Applications*, vol. 169, no. 8, 2017, pp. 32–35., doi:10.5120/ijca2017914816;

- [15] Breiman, L., 2001. Random Forests – Statistics Department. University of California – Berkeley. Disponível em: [www.stat.berkeley.edu/~breiman/randomforest2001.pdf](http://www.stat.berkeley.edu/~breiman/randomforest2001.pdf);
- [16] Friedman J, Hastie T, Tibshirani R, et al. (2000). “Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors).” *The annals of statistics*, 28(2), 337–407;
- [17] Natekin, Alexey, and Alois Knoll. “Gradient Boosting Machines, a Tutorial.” *Frontiers in Neurorobotics*, Frontiers Media S.A., 2013, [www.ncbi.nlm.nih.gov/pmc/articles/PMC3885826/](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3885826/);
- [18] Srivastava D. K., Bhambhu L. “Data classification using support vector machine”. *Journal of Theoretical and Applied Information Technology*, 2009;
- [19] D. Arthur and S. Vassilvitskii, “K-Means++: the Advantages of Careful Seeding,” *Proc. eighteenth Annu. ACM-SIAM Symp. Discret. algorithms*, pp. 1027–1025, 2007.
- [20] CHEN, T.; GUESTRIN, C. XGBoost: A scalable tree boosting system. Working Paper, 2016.
- [21] FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*, v. 29, n. 5, p. 1189-1232, 2001.
- [22] HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. (2008). *The Elements of Statistical Learning* (2nd ed.). Springer. ISBN 0-387-95284-5.