UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Lucas Yudi Matsuhashi

Predição de escalação de jogadores no jogo Cartola FC utilizando aprendizado de máquina

Uberlândia, Brasil

2025

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Lucas Yudi Matsuhashi

Predição de escalação de jogadores no jogo Cartola FC utilizando aprendizado de máquina

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Ciências da Computação.

Orientador: Paulo Henrique Ribeiro Gabriel

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciências da Computação

Uberlândia, Brasil

2025

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Faculdade de Computação
Av. João Naves de Ávila, nº 2121, Bloco 1A - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4144 - http://www.portal.facom.ufu.br/ facom@ufu.br

ATA DE DEFESA - GRADUAÇÃO

Curso de Graduação em:	Ciência da Computação: Bacharelado				
Defesa de:	Projeto de Graduação 2 (GBC082)				
Data:	14/05/2025	Hora de início:	10:05	Hora de encerramento:	11:35
Matrícula do Discente:	11711BCC031				
Nome do Discente:	Lucas Yudi Matsuhashi				
Título do Trabalho:	Predição de escalação de jogadores no jogo Cartola FC utilizando aprendizado de máquina				
A carga horária curricular foi cumprida integralmente?	(X) Sim ( ) Não				

Reuniu-se remotamente através da plataforma MS Teams, da Universidade Federal de Uberlândia, a Banca Examinadora, designada pelo Colegiado do Curso de Graduação em Ciência da Computação, assim composta: Professores: Dra. Fernanda Maria da Cunha Santos - FACOM/UFU; Dr. Rodrigo Sanches Miani - FACOM/UFU; e Dr. Paulo Henrique Ribeiro Gabriel - FACOM/UFU, orientador do candidato.

Iniciando os trabalhos, o presidente da mesa, Dr. Paulo Henrique Ribeiro Gabriel, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao discente a palavra, para a exposição do seu trabalho. A duração da apresentação do discente e o tempo de arguição e resposta foram conforme as normas do curso.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

(X) Aprovado Nota 90

OU

( ) Aprovado(a) sem nota.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Paulo Henrique Ribeiro Gabriel**, **Professor(a) do**

Magistério Superior, em 14/05/2025, às 11:36, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Rodrigo Sanches Miani**, **Professor(a) do Magistério Superior**, em 14/05/2025, às 11:37, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Fernanda Maria da Cunha Santos**, **Professor(a) do Magistério Superior**, em 14/05/2025, às 11:37, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **6333837** e o código CRC **CF4DE7F1**.

Referência: Processo nº 23117.031636/2025-58 SEI nº 6333837

Resumo

Este trabalho propõe o uso de algoritmos de aprendizado de máquina e otimização para prever o desempenho de jogadores de futebol no *fantasy game* Cartola FC, com o objetivo de montar equipes otimizadas para maximizar a pontuação ao longo do campeonato. Foi construída uma base de dados composta por informações históricas de desempenho individual dos atletas, juntamente com informações contextuais das partidas. Após a sua definição, os dados passaram pela etapa de pré-processamento, que envolve normalização, seleção de atributos e balanceamento das bases. Esses dados foram usados como entrada para diferentes modelos preditivos cuja sua saída consiste na pontuação que cada atleta alcançará em determinada rodada. Foram explorados os seguintes métodos: Máquina de Vetores de Suporte (SVM), Floresta Aleatória, Catboost e uma Rede Neural Recorrente (RNR). Em seguida, foi aplicado um método baseado no algoritmo genético para criar a melhor configuração de time, levando em consideração as restrições de orçamento e formação tática, retornando a escalação mais adequada. A abordagem proposta, de predição e otimização, mostrou-se eficaz, atingindo a média de 91.01 pontos ao longo do período proposto, com a pontuação máxima de 113.96 pontos na vigésima nona rodada do campeonato.

Palavras-chave: Cartola FC, Aprendizado de Máquina, Regressão Supervisionado, Modelos Preditivos e Algoritmo Genético.

Abstract

This work proposes using machine learning and optimization algorithms to predict the performance of football players in the fantasy game Cartola FC, aiming to assemble optimized teams to maximize the total score throughout the championship. We constructed a dataset using historical information on individual player performance and contextual data from the matches. After its construction, the dataset underwent a preprocessing phase that involved normalization, feature selection, and data balancing. These processed data were used as input for different predictive models, whose output corresponds to the expected score of each player in a given round. We explore the following methods: Support Vector Machine (SVM), Random Forest, CatBoost, and a Recurrent Neural Network (RNN). Subsequently, a method based on a genetic algorithm was applied to determine the best team configuration, considering budget constraints and tactical formations, ultimately returning the most suitable lineup. The proposed prediction and optimization approach proved effective, achieving an average of 91.01 points over the evaluated period, with a maximum score of 113.96 points in the twenty-ninth round of the championship.

Keywords: Cartola FC, machine learning, supervised regression, predictive models, genetic algorithm.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

Segundo Mueller e Massaron (2021), aprendizado de máquina é um campo da Inteligência Artificial que possibilita que um sistema possa aprender a partir de dados e não por meio de programação, possibilitando extrair novos conhecimentos a partir de grandes volumes de dados, podendo ser aplicado em diversas áreas, incluindo esportes e jogos.

Em relação a esportes, o futebol ocupa a posição de esporte mais popular do Brasil, impulsionado por um histórico de importantes resultados, como os cinco títulos de Copa do Mundo e a revelação de grandes jogadores. Essa trajetória contribui para que o Campeonato Brasileiro de Futebol se destaque como um dos torneios mais importantes do cenário internacional. De fato, o interesse da população por esse torneio aumenta a cada ano: em 2023, o Campeonato Brasileiro bateu recordes tanto na audiência (VAQUER, 2023) como em público nos estádios (MANIAUDET; SILVA, 2023). Com relação a aplicação de meios de inteligência artificial em futebol, o (FOOTHUB, 2021) diz que existem exemplos de monitoramento de jogadores para avaliar o condicionamento físico dos jogadores e análises de pontos fortes e fracos dos times adversários.

Visando ser uma forma de entregar um novo tipo de lazer no esporte, foi criado em 2005 o Cartola FC (GLOBO, 2024), um jogo do estilo *fantasy game*, onde é possível simular a escalação de um time, composto por jogadores reais do Campeonato Brasileiro Série A de Futebol. O participante do jogo deverá escalar um time, formado por 11 atletas e um técnico, respeitando o limite financeiro. A pontuação é disputada a cada rodada, de acordo com o desempenho estatístico dos jogadores em partidas reais, através de um sistema conhecido como *scouts*. Gradualmente, o jogo conquistou notoriedade, alcançando em 2023 a marca de 5,8 milhões de times montados. Além disso, o aplicativo do Cartola FC figurou entre os mais baixados na categoria esporte na Play Store e Apple Store, como citado no (TREMONTI, 2023). Assim, por utilizar dados estatísticos para avaliação dos atletas, o Cartola FC pode ser um bom estudo de caso para aplicação de métodos de aprendizado de máquina.

Desse modo, o objetivo deste trabalho é apresentar alguns métodos para predição de escalação de jogadores no jogo Cartola FC e realizar comparações quanto aos resultados obtidos, respeitando a questão financeira e os esquemas táticos presentes no jogo, de tal forma que possamos observar, estatisticamente, o desempenho dos atletas ao longo do campeonato. Também destaca-se a atualização de alguns trabalhos passados, justificado pela adição de novos aspectos no jogo, como o capitão, onde um determinado jogador ganha possibilidade de receber multiplicador de pontuação, e o banco de reservas, uma

forma de contornar a situação de um determinado jogador não entrar em campo numa determinada rodada, fazendo com que ele não receba pontuação.

Para isso, foi coletado os dados referente ao jogo de diversas fontes, como GitHub, portal Kaggle e do próprio site do Cartola FC, utilizando uma API. Após a coleta, os dados passaram por pré-processamento, como limpeza e remoção de dados, inclusão de novas variáveis e eliminação de *outliers*.

Em sequência, foram criados modelos preditivos para avaliar o desempenho dos atletas ao longo do período previsto, com o uso de algoritmos de aprendizado de máquina, estipulado em 19 rodadas do segundo turno do campeonato. Após isso, os dados passaram pelo processo de importância de atributos, no qual consiste em remover determinadas variáveis que são considerados menos importantes para o modelo. Assim, obtido as melhores variáveis, novos modelos foram criado, repetindo o processo passado.

Obtido os desempenhos dos atletas, foi avaliado o melhor esquema tático para a escalação do time, juntamente com a escolha dos seus jogadores, utilizando conceitos de algoritmos genéticos. Por fim, foi escolhido o capitão, um multiplicador de pontuação que é escolhido dentro da equipe, assim como o banco de reserva, que consiste numa lista de atletas que entram no time caso algum dos atletas escolhidos do time titular não participe do seu jogo.

2 Fundamentação Teórica

Neste capítulo será apresentado os fundamentos teóricos com relação ao tema, que irão servir de base para compreender o contexto do projeto.

2.1 Cartola FC

Cartola FC é um jogo eletrônico que simula o campeonato brasileiro de futebol no estilo *fantasy game*, no qual os usuários escalam times semelhantes ao futebol, ou seja, composto por onze jogadores e um técnico, que disputam o campeonato na vida real (GLOBO, 2024). Estes jogadores geram pontuações a cada rodada e o total de pontos dos atletas e técnico correspondem a pontuação da rodada. O objetivo final dos usuários é obter o máximo de pontuação total ao longo dos 38 rodadas do campeonato.

No começo do campeonato, os usuários recebem uma certa quantia em moeda virtual do jogo, denominado "cartoletas", para escalar o seu time, composto por onze atletas e um técnico. Esses atletas, ao longo do campeonato, vão receber valorização ou desvalorização no seu preço, conforme o seu valor atual e a pontuação que ele alcançar numa determinada rodada.

A escalação dos atletas é distribuída conforme a sua posição, que pode ser goleiro, zagueiro, lateral, meio-campista, atacante ou técnico. Com eles, o usuário irá escalar o seu time conforme o esquema tático escolhido. Por exemplo, caso o esquema seja 4-3-3, o usuário irá escalar 1 goleiro, 2 zagueiros, 2 laterias, 3 meio-campistas, 3 atacantes e 1 técnico.

A pontuação de um atleta é resultado do seu desempenho em campo, que é medido utilizando dados estatísticos, nomeado como *scout*, que podem ser positivos ou negativos. Além disso, o atleta irá sofrer valorização ou desvalorização conforme a sua pontuação e valor original. Caso o atleta tenha um baixo custo, irá precisar de pouca pontuação para valorizar, enquanto o atleta de alto custo terá que pontuar bem para valorizar. Os *scouts* sofreram várias mudanças ao longo dos anos, sendo criados, retirados ou alterados. Nas tabelas 1 e 2 são mostrados os *scouts* definidos para o ano de 2023.

Nota-se que os *scouts* de defesa e defesa de pênalti são exclusivo para goleiros, enquanto o *scout* de jogo sem sofrer gols pode ser aplicado somente aos goleiros, zagueiros e laterais. Já a pontuação de vitória é somente para técnico. Além disso, a pontuação do técnico é peculiar em relação aos demais atletas, pois corresponde à média da pontuação dos jogadores da sua equipe, podendo ser acrescentado 1 ponto caso o seu time saia vencedor da partida.

Tabela 1 – *Scouts* positivos para o ano de 2023.

Gol	+8.00
Assistência	+5.00
Finalização na Trave	+3.00
Finalização Defendida	+1.20
Finalização para Fora	+0.80
Falta Sofrida	+0.50
Pênalti Sofrido	+1.00
Defesa de Pênalti	+7.00
Jogo sem Sofrer Gols	+5.00
Defesa	+1.00
Desarme	+1.20
Vitória	+1.00

Tabela 2 – *Scouts* negativos para o ano de 2023.

Pênalti Perdido	-4.00
Impedimento	-0.10
Gol Contra	-3.00
Cartão Amarelo	-1.00
Cartão Vermelho	-3.00
Gol Sofrido	-1.00
Falta Cometida	-0.30
Pênalti Cometido	-1.00

Por exemplo, na primeira rodada de 2023, o atleta Cano, do Fluminense, fez 1 gol, 1 finalização para fora, 1 finalização na trave, 1 desarme, 2 impedimentos, e 1 pênalti perdido. Ao final, a pontuação dele foi $1 \cdot (8.0) + 1 \cdot (0.8) + 1 \cdot (3.0) + 1 \cdot (1.2) + 2 \cdot (-0.1) + 1 \cdot (-4.0) = 8.8$ pontos. Contudo, ele sofreu uma desvalorização de 2.13 cartoletas, pois o seu valor inicial era de 22 cartoletas, considerado alto.

Alguns detalhes foram adicionados ao jogo no decorrer dos anos. Para evitar que o usuário não receba pontuação por um determinado atleta não entrar em campo, o Cartola FC instituiu o banco de reserva, no qual, para cada posição, o usuário estabelece um atleta para servir como troca quando o atleta titular não participar do jogo, recebendo a pontuação que ele alcançar na partida. Esse atleta reserva é necessário que o seu valor seja menor que os atletas titulares da sua posição.

Ao final da sua escalação, o usuário escolhe um atleta, com exceção do técnico, para ser denominado como capitão da sua equipe, ao qual dará o bônus de 1.5 vezes a sua pontuação final. No caso do atleta Cano, citado anteriormente, caso ele estivesse como capitão, a equipe receberia 4.4 pontos a mais pelo seu desempenho. Naturalmente, caso o capitão receba uma nota negativa, o acréscimo de nota que o time irá receber também será negativa.

2.2 Aprendizado de Máquina

Aprendizado de máquina é uma área da Inteligência Artificial que permite que um sistema possa aprender padrões a partir de dados, ao invés de programação explícita, sejam elas usadas para análise de um determinado aspecto ou predição de resultados, como dito no (MUELLER; MASSARON, 2021).

De acordo com Mitchell (1997), a história de aprendizado de máquina é datada na década de 1950, quando foram criados os primeiros algoritmos de autoaprendizado para jogar damas. Porém, o seu real desenvolvimento começou nos anos de 1990, quando obteve desenvolvimento suficiente na área de estatística e da teoria da probabilidade para dar suporte teórico aos seus algoritmos, juntamente com o aumento da disponibilidade de informações digitalizadas disponíveis via internet. Além disso, a melhoria nos sistemas distribuídos nas últimas décadas motivaram a utilização de grande quantidade de dados para que possam ser processadas, aumentando gradativamente o uso de algoritmos de inteligência artificial.

Conforme apresentado por Gala (2023), os algoritmos de aprendizado de máquina são classificados em três tipos. O aprendizado supervisionado treina um conjunto de dados rotulados, ou seja, os resultados são pré-definidos e o algoritmo tenta encontrar padrões entre os rótulos e os resultados. Já o aprendizado não supervisionado treina um conjunto de dados sem rótulos, identificando padrões e estruturas nos dados sem ter exemplos de saídas desejadas. Finalmente, o aprendizado por reforço usa-se um agente para auxiliar no treinamento dos dados, interagindo dinamicamente com o ambiente, de forma que recebe *feedback* em forma de recompensas ou penalidades que o incentiva a aprender determinadas ações.

O processo de aprendizado de máquina consiste em coleta de dados a respeito do tema, limpeza dos dados coletados para facilitar o processo de aprendizado, criação e treinamento de um modelo para reconhecer os padrões dos dados e prever os resultados com o modelo criado. Diversos modelos (algoritmos) de aprendizado de máquina podem ser explorados. Neste trabalho, foram implementados os métodos descritos a seguir.

2.2.1 Máquina de Vetores de Suporte

O algoritmo máquina de vetores de suporte (SVM, do inglês *Support Vector Machine*) é uma técnica de aprendizado supervisionado que visa encontrar o hiperplano ótimo que separa as classes de dados no espaço de características. Inicialmente projetada para a classificação binária de objetos, o SVM foi estendido para regressão por meio da variante conhecida como *Support Vector Regression* (SVR). Seu objetivo é encontrar uma função que tenha, no máximo, uma determinada margem de tolerância ε com respeito aos valores reais dos dados de treinamento, e ao mesmo tempo seja a mais plana possível (DRUCKER

et al., 1997).

No caso da regressão, Smola e Schölkopf (2004) explica que o SVM procura minimizar uma função de perda baseada na insensibilidade ao erro (ε-*insensitive loss function*), o que permite que erros dentro da margem ε não sejam penalizados. Isso confere ao modelo maior robustez frente a pequenas flutuações nos dados. O problema de otimização do SVR pode ser formulado como um problema de programação convexa, resolvido geralmente por métodos de programação quadrática, garantindo a convergência para a solução global.

Uma das principais vantagens do SVM está na sua efetividade quanto a aplicação em altas dimensões por meio do uso de funções chamadas de *kernels*, que possibilitam mapear os dados de entrada para um espaço de características de alta dimensão, adaptando diversos padrões complexos sem que seja necessário especificar explicitamente uma transformação dos dados (IBM, 2023).

Além disso, o SVM tende a sofrer menos de *overfitting* do modelo, especialmente quando o tamanho das amostras é pequeno em comparação ao número de atributos. Isso ocorre em virtude do uso do princípio da margem máxima, que favorece soluções mais generalizáveis. No entanto, Cortes e Vapnik (1995) afirma que o algoritmo apresenta sensibilidade em relação à escolha dos hiper-parâmetros (como ε e os parâmetros do *kernel*) e o alto custo computacional em conjuntos de dados muito grandes podem limitar sua aplicabilidade em alguns contextos.

2.2.2 Floresta Aleatória

O algoritmo Floresta Aleatória (RF, do inglês *Random Forest*) é uma técnica de aprendizado de máquina baseada em árvores de decisão, pertencente à classe dos métodos de *ensemble learning*. Desenvolvido por Breiman (2001), a Floresta Aleatória busca superar as limitações das árvores de decisão individuais – que são altamente sensíveis a variações nos dados – por meio da agregação de múltiplas árvores construídas de forma independente. Essa abordagem reduz significativamente o risco de *overfitting* e melhora a generalização do modelo. A construção de cada árvore ocorre sobre diferentes subconjuntos dos dados, amostrados com reposição (técnica de *bootstrap*), enquanto a seleção de atributos em cada divisão da árvore é feita de forma aleatória, promovendo a diversidade entre os modelos base.

Durante o treinamento, o algoritmo constrói um grande número de árvores de decisão, cada uma aprendendo padrões específicos do conjunto de dados. Para problemas de regressão, a predição final do Floresta Aleatória é obtida por meio da média das predições individuais das árvores, enquanto em tarefas de classificação, utiliza-se o voto majoritário. A combinação de múltiplas árvores contribui para a redução da variância do modelo e maior robustez frente a ruídos e *outliers*, característica que o torna especialmente

eficaz em conjuntos de dados de alta dimensionalidade ou com presença de atributos colineares.

Segundo consta na biblioteca scikit-learn (2025b), um dos principais atrativos do Floresta Aleatória é sua capacidade de fornecer medidas de importância das variáveis. Isso é realizado ao observar a redução da impureza (como o índice de Gini ou entropia) provocada por cada variável em todas as árvores. Essa métrica permite a interpretação e seleção de atributos mais relevantes, o que auxilia na compreensão do problema e na construção de modelos mais simples e eficientes. Além disso, técnicas como o Permutação de Importância de Atributos também são comumente aplicadas ao Floresta Aleatória, proporcionando uma visão mais estável e imparcial da importância dos atributos.

No entanto, o Floresta Aleatória apresenta algumas limitações, como o custo computacional elevado para conjuntos de dados muito grandes ou com alto número de árvores, e a dificuldade de interpretação detalhada do modelo como um todo, o que pode ser um obstáculo em aplicações que exigem transparência. Ainda assim, sua eficácia empírica é amplamente reconhecida, sendo frequentemente utilizada como *benchmark* em tarefas de classificação e regressão.

Diversos estudos destacam o desempenho do Floresta Aleatória em diferentes domínios. Segundo Breiman (2001), o método alcança alta acurácia em problemas supervisionados sem a necessidade de ajustes extensivos de parâmetros. Além disso, trabalhos posteriores demonstram sua estabilidade mesmo com dados ruidosos e desbalanceados (LOUPPE, 2014).

2.2.3 Catboost

Como dito no site oficial[1], o CatBoost (*Categorical Boosting*) é um algoritmo de aprendizado de máquina baseado em *gradient boosting* que foi desenvolvido com o objetivo de lidar de forma eficiente com variáveis categóricas e reduzir o *overfitting* em modelos de *boosting*.

Uma das principais contribuições do CatBoost está em seu tratamento nativo de variáveis categóricas, dispensando a necessidade de pré-processamentos como *one-hot encoding* ou *label encoding*. Em vez disso, o algoritmo utiliza uma técnica baseada em estatísticas de substituição com permutação ordenada, o que permite representar categorias de forma mais robusta durante o treinamento. Essa abordagem ajuda a evitar a introdução de viés e melhora a capacidade do modelo de generalizar bem para novos dados.

Além disso, Prokhorenkova et al. (2018) expressam que o CatBoost introduz um novo método de cálculo de gradientes chamado Ordered Boosting, que visa mitigar o

[1] <https://catboost.ai/docs/en/>

problema de previsão com vazamento de informação (*target leakage*) comum em algoritmos de *boosting* tradicionais. Essa técnica simula o processo de treinamento como se os dados fossem processados de forma online, isto é, fazendo com que as estimativas para cada instância sejam calculadas com base apenas em observações anteriores. Isso permite um treinamento mais estável, especialmente em conjuntos de dados pequenos ou com colunas altamente informativas.

Prokhorenkova et al. (2018) ainda argumentam que o CatBoost também é reconhecido por sua eficiência computacional. Isso inclui otimizações em nível de CPU e GPU, além de paralelismo em múltiplos níveis do processo de *boosting*. Outra vantagem significativa do CatBoost é a sua robustez aos parâmetros de configuração: diferentemente de outros algoritmos, ele pode produzir bons resultados com poucos ajustes manuais, sendo altamente competitivo em tarefas de regressão, classificação e ordenação.

2.2.4 Redes Neurais Recorrentes

As Redes Neurais Recorrentes (RNNs) constituem uma classe de modelos de redes neurais especialmente desenvolvidos para o processamento de dados sequenciais ou temporais, como séries temporais, linguagem natural e sinais sensoriais. Diferentemente das redes neurais *feedforward* tradicionais, que assumem independência entre as entradas, as RNNs incorporam conexões recorrentes que permitem a retenção de informações de entradas anteriores na rede. Essa característica torna as RNNs adequadas para modelar relações dinâmicas e dependências temporais complexas (GOODFELLOW; BENGIO; COURVILLE, 2016).

O funcionamento de uma RNN baseia-se na manutenção de um estado oculto que é atualizado a cada passo temporal com base na entrada atual e no estado anterior. Formalmente, dado um vetor de entrada x_t e um vetor de estado oculto anterior h_{t-1}, o novo estado oculto h_t é calculado como

$$h_t = \sigma(Wx_1 + Uh_{t-1} + b),$$

onde W, U e b são parâmetros treináveis da rede e σ é uma função de ativação não linear (como as funções tangente hiperbólica, ou simplesmente tanh e unidade linear retificada, ou ReLU). O estado oculto permite que a rede aprenda a representar sequências com estrutura temporal, ainda que com limitações em sequências longas.

O treinamento de RNNs geralmente envolve grandes volumes de dados e considerável poder computacional, principalmente em tarefas com sequências extensas e vocabulários grandes, como em PLN. Estratégias como *teacher forcing*, *dropout* e inicialização adequada dos pesos são comumente adotadas para estabilizar e acelerar o treinamento. Em contextos práticos, as RNNs são frequentemente substituídas por arquiteturas como

transformadores, embora ainda desempenhem papel relevante em aplicações com restrições de recursos ou que demandem previsões rápidas em tempo real.

2.3 Algoritmos Genéticos

Algoritmos genéticos são métodos de busca e otimização caracterizados por aproveitar conceitos de biologia e evolução, como a mutação, hereditariedade e recombinação (*crossover*) (EIBEN; SMITH, 2015).

A complexidade computacional do problema no qual os algoritmos genéticos são implementados geralmente são altas e, portanto, computacionalmente, não é viável propor uma solução exata e perfeita. Sendo assim, a ideia dos algoritmos genéticos é sugerir uma boa resposta suficiente que satisfaz como solução do problema.

A estratégia usada nos algoritmos genéticos é de uma busca paralela e coordenada, que são direcionadas de acordo com os resultados de uma função-objetivo, visando minimizá-la ou maximizá-la, explorando as informações históricas para determinar os novos pontos de busca com melhores resultados. Assim, o processo desses algoritmos são iterativos.

Para cada iteração, o algoritmo determina os processos de seleção e reprodução a partir de um grupo escolhido chamado de população, que são seguidos de avaliações por meio de função-objetivo e aplicação de mutação e *crossover* (EIBEN; SMITH, 2015).

Segundo Goldberg (1989), a principal vantagem dos AGs reside em sua capacidade de escapar de ótimos locais, explorando eficientemente espaços de busca grandes e não lineares. Diferentemente de métodos determinísticos, como gradiente descendente, os AGs não requerem conhecimento prévio sobre a derivabilidade da função objetivo, sendo particularmente úteis em contextos com múltiplos parâmetros e restrições complexas. No entanto, a eficácia do algoritmo depende criticamente da escolha de parâmetros como tamanho da população, taxas de cruzamento e mutação, além da definição apropriada da função de *fitness*.

2.4 Trabalhos Relacionados

Ribeiro (2019) propôs um modelo para predição de escalações de times no Cartola FC, utilizando métodos de pré-processamento e análise de dados, principalmente com foco em visualização por boxplot, com o uso de redes neurais perceptron de múltiplas camadas para classificar os atletas. Destaca-se a modelagem formal do processo de escalação sendo tratado como um problema de otimização, definindo a função objetivo e restrições dentro do modelo. Por fim, é realizado a modelagem e implementação do algoritmo genético para escalação do time. A conclusão obtida foi que conforme o aumento do limite de patrimô-

nio disponível para escalação dos atletas influenciam na qualidade dos times escalados, juntamente com a contribuição matemática do processo de escalação de atletas como um problema de otimização.

Coelho Filho (2021) utilizou de análise descritiva das variáveis, como histogramas e diagramas de dispersão para entender os dados. Contudo, o foco do autor é usar os modelos de equações de estimação generalizadas para criação dos modelos para predizer a pontuação dos atletas. É interessante notar as informações estatísticas que o autor traz, juntamente com a criação de novas variáveis para ajudar a explicar os dados e garantir melhor qualidade para o resultado. Um dos fatores importantes citados na conclusão é que nem sempre o modelo mais adequado, na teoria, é o que apresentará o melhor resultado. Contudo, um grande foco em estatística pode trazer uma boa ideia para o nosso projeto.

Viscondi, Justo e García (2017) sugerem a aplicação de métodos de aprendizado de máquina para predição de escalação do time com o intuito de atingir a melhor pontuação no Cartola FC, com foco em reduzir o risco de escalação de jogadores, criando grupos de atletas para que seja maximizada a pontuação. Como resultado, o autor entende que não foi satisfatório quanto aos resultados da predição, porém, foram positivos quanto ao uso da clusterização dos atletas no pré-processamento. É importante notar que este artigo se relaciona bastante com o nosso trabalho, pois as propostas e o objetivo se assemelham bastante. Contudo, desde 2017, foram aplicados diversas mudanças no jogo, como adição do "capitão"e "banco de reservas"e mudança em diversos *scouts*. Assim, visa-se a necessidade de atualizar o trabalho, além de trazer novos conceitos.

No trabalho de Cruz, Sousa e Calçada (2020) foram realizadas a validação de um modelo de perfil de jogador para a escalação no Cartola FC, focando exclusivamente na posição de goleiro e atacante, utilizando técnicas de Redes de Associação Filtradas como meio de estruturar e compreender os padrões dos dados e Inteligência Artificial Explicável para criar os modelos. Um ponto diferente deste artigo é a categorização dos atletas por atributos conforme os pontos conquistados por rodada, semelhante aos quartis. Como resultado, os autores identificam um padrão de determinados atributos na classe de jogadores considerados bons, como defesa difícil nos goleiros e gols, assistências e roubadas de bola para os atacantes. Contudo, a árvore de decisão gerado não possibilitou uma identificação dos fatores importantes que são diretamente conectados aos jogadores que alcançam maiores pontuações. devido a uma pobreza do conhecimento gerado e a dificuldade na sua análise, devido ao seu tamanho. Por final, esse artigo apresenta uma proposta de organização dos dados utilizando regras de associação que pode trazer uma boa ideia para o nosso trabalho.

3 Desenvolvimento

Neste capítulo, é apresentado o desenvolvimento do deste trabalho, desde a coleta dos dados até a elaboração dos times que disputaram as rodadas ao longo do campeonato.

3.1 Coleta pré-processamento dos dados

A primeira etapa consiste na coleta dos dados que serão utilizados neste trabalho. Para tanto, serão usados os dados de alguns meios para a realização do presente trabalho. Primeiramente, foi usado conjunto de dados de um repositório de GitHub chamado "caRtola" (GOMIDE, 2023), no qual é possível obter informações de todos os atletas participantes do campeonato no período de 2014 a 2024. Além disso, algumas informações foram obtidas do próprio portal do Cartola FC, com o uso de um API, para auxiliar no desenvolvimento do trabalho. Por fim, foram obtidos dados correspondentes aos placares dos confrontos ao longo do campeonato do Kaggle (DUQUE, 2024).

Após a coleta, todos os dados passaram por um processo de pré-processamento, o qual consiste, principalmente, em remover ou consertar os dados falhos e criar novos atributos (*features*).

Além disso, foi realizada a unificação das informações contidas nas três fontes de dados em um único conjunto de dados, onde, para cada jogador, foram inseridas informações individuais e coletivas, que serão apresentados em sequência. Desse modo, após o processo de pré-processamento, as colunas do conjunto de dados, juntamente com o seu tipo de dado, são seguinte:

- atleta_id: Identificação do atleta. Valor inteiro positivo.
- clube_id: Identificação do time do atleta. Valor inteiro positivo.
- posição_id: Identificação da posição do atleta. Valor inteiro, que varia entre [1, 6].
- rodada: Rodada que as informações foram alcançadas. Valor inteiro que varia entre [1, 38].
- preco: Preço dos atletas na rodada. Valor flutuante positiva.
- num_pontos: Pontuação obtida na rodada. Valor flutuante.
- num_jogos: Número de jogos que o atleta participou. Valor inteiro positivo.
- variacao: variação do preço do jogador em relação ao início da rodada; Valor flutuante.

- media: Média da pontuação (num_pontos) obtida ao longo do campeonato. Valor flutuante.
- status: Status do jogador, conforme a sua disponibilidade de participar da rodada. Valor inteiro que varia entre [1, 7].
- A: Número de assistências. Valor inteiro não negativo.
- CA: Número de cartões amarelos. Valor inteiro não negativo.
- CV: Número de cartões vermelhos. Valor inteiro não negativo.
- DE: Número de defesas. Valor inteiro não negativo.
- DS: Número de desarmes. Valor inteiro não negativo.
- DP: Número de defesas de pênaltis. Valor inteiro não negativo.
- FC: Número de faltas cometidas. Valor inteiro não negativo.
- FD: Número de finalizações para o gol. Valor inteiro não negativo.
- FF: Número de finalizações para fora do gol. Valor inteiro não negativo.
- FS: Número de faltas sofridas. Valor inteiro não negativo.
- FT: Número de finalizações na trave. Valor inteiro não negativo.
- G: Número de gols. Valor inteiro não negativo.
- GC: Número de gols contras. Valor inteiro não negativo.
- I: Número de impedimentos. Valor inteiro não negativo.
- PC: Número de pênaltis cometidos. Valor inteiro não negativo.
- PP: Número de pênaltis perdidos. Valor inteiro não negativo.
- PS: Número de pênaltis sofridos. Valor inteiro não negativo.
- SG: Número de bônus de saldo de gols. Valor inteiro não negativo.
- V: Quantidade de vitórias obtidas pelo time até a rodada. Valor inteiro não negativo.
- jogou: Informação se o jogador participou do jogo naquela rodada. Valor booleano.
- media_5: Média de pontuação dos últimos 5 jogos que o atleta participou. Valor flutuante.
- adversario: Identificação do adversário da rodada. Valor inteiro positivo.

- prox_adversario: Identificação do adversário da próxima rodada. Valor inteiro positivo.
- posicao: Posição do time na tabela. Valor inteiro que varia entre [1, 20].
- sequencia_marcando: Sequência de rodadas que o time está marcando. Caso esteja numa sequência de jogos marcando gols, o valor será positivo. Caso contrário, será negativo. Valor inteiro.
- sequencia_SG: Sequência de rodadas que o time está sofrendo gols. Caso esteja numa sequência de jogos sofrendo gols, o valor será positivo. Caso contrário, será negativo. Valor inteiro.
- resultados_5: média de pontos do campeonato obtidos nos últimos 5 jogos. Valor flutuante que varia entre [0, 3].

Dessa forma, as informações que cada atleta carrega por rodada podem ser dividida em alguns tipos:

- Informações pessoais: Dados que correspondem a identificação do atleta. São deste grupo atleta_id, clube_id, posição_id, rodada, preco, media, variacao e status.
- *Scouts*: São informações coletadas ao longo da partida pelos organizadores do Cartola FC, no qual se resume à pontuação final do atleta na rodada. São deste grupo A, CA, CV, DE, DS, DP, FC, FD, FF, FS, FT, G, GC, GS, I, PC, PP, PS, SG, jogou e num_pontos.
- Informações adicionais: Dados que foram inseridas através do processo de pré-processamento. São deste grupo media_5, adversaio, prox_adversario, mando, posicao, prox_mando, sequencia_marcando, sequencia_SG, resultados_5.

Todos os atletas participantes têm dados em relação às informações pessoais e adicionais. Contudo, apenas o técnico, por ser uma posição que alcança a sua pontuação ao realizar a média dos demais atletas do time, não terá pontuações alcançadas via *scouts*. Assim, todos os valores presentes neste grupo será zerado em relação a posição de técnico.

Alguns *scouts* são exclusivos para determinadas posições. A coluna SG (Bônus de Saldo de Gols) é um *scout* específico para os defensores, ou seja, goleiros, laterais e zagueiros (isto é, posição_id igual a 1, 2 ou 3). Já os *scouts* de DE, DP e GS são exclusivos de goleiros (posição_id igual a 1).

Além do conjunto de dados principal, foram utilizados alguns conjunto de dados adicionais para facilitar o desenvolvimento deste trabalho e não sobrecarregar o conjunto de dados principal, como a tabela de classificação dos times ao longo do campeonato,

uma relação entre a identificação dos jogadores com os seus nomes e uma relação dos confrontos do campeonato, juntamente com as informações sobre os times mandantes e visitantes e os placares atualizados.

Outras etapas de pré-processamento são a eliminação de dados de uma determinada rodada do campeonato e o tratamento de *outliers*. Essas etapas são descritas a seguir.

3.1.1 Eliminação dos dados por rodada prevista

Como o objetivo deste trabalho é prever a melhor escalação para uma determinada rodada. No entanto, o conjunto de dados principal tem informações sobre todas as rodadas, de modo que os dados correspondentes às rodadas posteriores a rodada selecionada devem ser removidos. Por exemplo: se a predição for considerada na rodada 30, todas as informações presentes no conjunto de dados depois da rodada 30 será removida, sendo consideradas as informações das rodadas 1 ao 29 nos procedimentos posteriores.

Destaca-se que esse processo é necessário por este projeto estar sendo realizado considerando o campeonato que já foi finalizado. Caso a análise fosse feita com base em um campeonato corrente, com as rodadas sendo realizados em tempo real, essa operação não seria necessária. Contudo, ainda é necessário observar os atletas que estarão disponíveis para as suas respectivas partidas. Este processo pode ser feito analisando o status de cada atleta, um dado oferecido pelo API do Cartola FC, sendo que o status 'Provável' sinaliza que o jogador provavelmente irá compor o time titular (ou ao menos participar da partida). Contudo, vale ressaltar que esta informação não é certa, pois como o prazo de escalação do time é até o mercado fechar, isto é, alguns minutos antes da primeira partida válida da rodada, não há segurança de que um determinado atleta dos jogos posteriores irá participar, dada as circunstâncias variadas que ele pode sofrer até a realização da partida, mesmo com o status de 'Provável'. Todos os dados coletados a partir do site oficial foi obtido de forma grátis, sem o uso de uma conta premiada.

3.1.2 Tratamento de *outliers*

Os *outliers* são dados que são considerados anormais ou fogem da normalidade em relação aos demais dados, podendo causar algum desvio nos resultados obtidos por meio de análises (AQUARELA, 2017). Para encontrar os *outliers*, é possível observar os dados diretamente utilizando tabelas ou planilhas de dados, ou partir para uma perspectiva estatística, como usar distribuição dos dados para detectar os pontos discrepantes, figuras para enxergar as anomalias visualmente ou realizar o processo de agrupamento dos dados.

Neste trabalho, algumas das pontuações obtidas pelos atletas ao longo do campeonato, em determinada rodada, podem ser muito superiores ou inferiores em relação às

demais pontuações, a ponto de serem considerados *outliers*.

A detecção das anomalias foi realizada via técnica estatística Z-score (ou escore padrão), uma medida simples e eficaz para os conjuntos que seguem uma distribuição normal (DATASCIENCESPHERE, 2024). Neste método, para cada dado, será medido uma distância em relação à média do conjunto de dados, traduzida como desvio padrão. Caso o valor obtido for um valor superior a um determinado limiar, esses dados serão considerados uma anomalia e, posteriormente, eles serão tratados.

O Z-score deste trabalho é indicado pela fórmula $Z = (X - \mu)/\sigma$, onde X representa a pontuação obtida naquela rodada (num_pontos), μ é a média da pontuação e σ, o desvio padrão das pontuações. O limiar escolhido para ser considerado *outlier* foi 3, ou seja, valores Z-scores acima de 3 ou inferiores a -3 foram considerados *outliers*.

As anomalias não foram removidas da base de dados, visto que, apesar de serem considerados exceção em comparação com os demais dados, elas são importantes para o processo preditivo, pois precisa deixar claro que um determinado atleta participou da partida, no qual teve um rendimento muito bom ou ruim. Para isso, os atletas que obtiveram notas acima do limite superior de outlier, isto é $\mu + 3Z$, conforme a fórmula citada, foram igualadas ao valor do limite superior, enquanto as pontuações abaixo das notas inferiores, isto é $\mu - 3Z$, foram igualadas ao valor do limite inferior.

3.2 Divisão do conjunto de dados

Após o pré-processamento, os dados foram divididos em dados de treino e teste. Os dados de treino são os dados que foram utilizados pelos algoritmos de aprendizado de máquina para criação dos modelos. Já os dados de teste são os dados fornecidos ao modelo para poder realizar simulações reais, possibilitando análises de desempenhos dos modelos criados. Além disso, os dados de treino e teste foram divididos conforme a característica de manipulação das variáveis. As variáveis independentes são manipuladas pelos agentes (no caso deste trabalho, o atleta), enquanto as variáveis dependentes são mensuradas como resposta das variações das variáveis independentes (LIMA, 2022).

Existem alguns métodos de separação dos dados. O mais simples é realizar o corte dos dados aleatoriamente em dados de treino e de teste, sendo que os dados de treino correspondem maiores tamanho em relação ao de teste (em torno de 70% dos dados para treino e 30% dos dados para teste). Contudo, os dados observados neste trabalho têm uma característica sequencial de tempo, de modo que a ordem dos dados é fundamental e as observações vizinhas são dependentes, sinalizando uma série temporal. Assim, essa forma de separar os dados não é a mais adequada, pois desconsidera a dependência dos dados vizinhos.

Entretanto, para os objetivos deste trabalho, a divisão é relativamente simples. O próprio processo de predizer os resultados da rodada, utilizando as informações das rodadas anteriores já oferece a ideia de divisão dos dados, isto é, os dados das rodadas anteriores correspondem à dados de treino e os dados da rodada atual, dados de teste.

Além disso, dentre os dados de treino e teste criados, a coluna 'num_pontos' for especificada como variável-alvo, ou seja, variável que é o foco do nosso projeto. Por final, os nossos dados foram divididos em X_train (atributos de treinamento), y_train (variável foco do treinamento), X_test (atributos de treinamento) e y_test (variável foco de teste).

3.3 Criação dos Modelos

Finalmente, foi realizada a criação dos modelos de aprendizado de máquina, descritos no capítulo anterior. Todos os algoritmos foram aplicados de bibliotecas conceituadas do cenário de aprendizado de máquina, para facilitar o processo de criação dos modelos, visualização dos resultados e manipulação dos hiper-parâmetros.

Floresta Aleatória: Para criar o modelo do algoritmo de Floresta Aleatória, foram utilizadas as funcionalidades presentes na biblioteca scikit-learn, denominado *RandomForestRegressor* (SCIKIT-LEARN, 2025b).

Máquina de Vetores de Suporte: O modelo do algoritmo de Máquina de Vetores de Suporte também utiliza as funcionalidades da biblioteca scikit-learn, denominado SVR (*Support Vector Regressor*) (SCIKIT-LEARN, 2025c).

CatBoost: O modelo do algoritmo de CatBoost utilizou as funcionalidades da biblioteca CatBoost do Python, denominado *CatBoostRegressor* (AI, 2025).

Rede Neural Recorrente: Finalmente, o modelo do algoritmo de Redes Neurais Recorrentes utilizou as funcionalidades da biblioteca *Keras*, utilizando apenas camadas classificadas como redes neurais simples (*SimpleRNN*) (KERAS, 2025).

Os algoritmos citados foram escolhidos de acordo com os tipos de algoritmos que existem para resolver um problema de regressão com dados temporais, no qual este trabalho é classificado. No algoritmo de floresta aleatória, é utilizado a técnica da árvore de decisão, enquanto na máquina de vetores de suporte é usado o hiperplano como suporte de regressão. Além disso, como exemplo de rede neural, temos o uso de rede neural recorrente. Por fim, o Catboost utiliza a técnica de *boosting*, apesar de também utilizar estruturas de árvores de decisão binária como base de predição (JEREMIAH, 2024).

3.4 Importância de Atributos

A Importância de Atributos é a técnica de avaliar o desempenho das variáveis dentro da predição. Para cada variável, é calculado a sua relação com a variável objetivo. a Importância de Atributos ajuda a simplificar modelos, possibilitando reduzir determinadas variáveis que pouco contribui, melhorando a eficiência em processamento e aumenta a compreensão do modelo (SHIN, 2023). Neste trabalho, foram utilizadas duas técnicas de Importância de Atributos, descritas a seguir.

Permutação de Importância de Atributos: Para o RF e SVM, é utilizada a Permutação de Importância de Atributos, que consiste embaralhar os valores de uma variável, mantendo as demais, e conferir o desempenho do modelo. Caso o resultado seja prejudicado, isto é, uma diferença no performance do modelo, significa que há uma relação considerável entre a variável embaralhada e o alvo e o grau de mudança indica o nível de contribuição da variável ao modelo (SCIKIT-LEARN, 2025a).

Prediction Value Change: Para o Catboost e RNN o método Prediction Value Change foi utilizado. Ela consiste em medir o quanto a previsão final varia se uma certa variável presente nos dados forem usadas para dividir os nós em uma árvore. É um método rápido, já que calcula a importância das variáveis de acordo com as estruturas das árvores já existentes, medindo a contribuição acumulada de cada variável na mudança dos valores previstos (CATBOOST, 2025).

Após o processo de criação dos modelos, é feito a escolha dos melhores atributos por meio da importância de atributos, de modo que foram escolhidos os 7 atributos mais importantes para cada modelo criado, de acordo com o algoritmo de aprendizado de máquina utilizado. Logo após a escolha, os modelos são recriados, baseados nos atributos selecionados.

3.5 Avaliação de Desempenho dos Modelos

Após a criação dos modelos dos respectivos algoritmos de aprendizado de máquina, foram criados métodos para analisar estatisticamente o desempenho de cada modelo. Utilizou-se três métricas para esta análise, para poder explicar melhor o desempenho do modelo.

3.5.1 Erro Quadrático Médio

O Erro Quadrático Médio (MSE) é o desvio padrão dos erros de previsão, ou seja, a distância entre a função de regressão, criada nos algoritmos, e os pontos de dados. Em

outras palavras, representa a concentração dos dados em torno da linha da função (ORACLE, 2025). Quanto menor o resultado obtido, mais perto o modelo está da realidade. A fórmula do MSE é dada por

$$MSE = \sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)}{n} \tag{3.1}$$

onde, n representa número de observações, y_i representa os valores observados, $\hat{y}_i$ representa os valores que foram preditos. Neste trabalho, o valor n representa o número de atletas, y_i os resultados reais das pontuações dos atletas na rodada de predição e $\hat{y}_i$, as pontuações que foram preditos pelos algoritmos.

Ao elevar ao quadrado a diferença entre o valor obtido e o valor real, o MSE sugere que é sensível aos erros, o que pode ser uma vantagem, por enfatizar erros, como também desvantagem, por estimar, de forma exagerada, um mau desempenho do modelo com presença de *outlier* (QUAL..., 2024).

3.5.2 Erro Médio Absoluto

O erro médio absoluto (MAE) é a média da diferença entre o valor real e o predito. Esta métrica não é sensível a *outliers* e verifica, somente, a diferença entre valores.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |(y_i - \hat{y}_i)| \tag{3.2}$$

As características das variáveis presentes na equação (3.2) são semelhantes ao Erro Quadrático Médio, mostrado na equação (3.1).

3.6 Escalação do time

Após predizer todos os atletas que vão disputar a rodada de predição, foi realizada a escalação do time, um problema de otimização, que visa escalar os atletas que obtiveram as melhores pontuações, de acordo com o limite financeiro proposto.

Para cada formação tática, os dados dos atletas, juntamente com a sua pontuação prevista, foram inseridos em função criada a partir de métodos de algoritmos genéticos. Inicialmente, foram criadas os indivíduos do algoritmo genético. Nesse caso, cada indivíduo é um time composto por 11 atletas e 1 técnico, formando a população base do algoritmo, respeitando a sua formação tática. Na figura 1, temos a configuração de um time com 1 goleiro, 2 laterais, 2 zagueiros, 3 meias, 3 atacantes e 1 técnico, conhecido como formação 4-3-3.

Posteriormente, em um processo iterativo determinado pelo tamanho da população, foram selecionados os melhores indivíduos, a partir da função-objetivo e método de

Figura 1 – Exemplo de time na formação tática 4-3-3.

seleção, e os determinou como pai, para criar novos filhos, por meio do *crossover* das informações do pai. Após este processo, os filhos criados podem passam por uma mutação, que é a substituição de um atleta por outro da mesma posição. No final, os novos filhos farão parte da nova população.

Em seguida, os indivíduos da nova população irão passar pelo processo de avaliação por meio da função objetivo. Essa função consiste em coletar o custo em cartoletas do time, juntamente com a soma de pontuação prevista de cada atleta. Caso o custo total do time esteja acima do limite proposto, a função retorna 0, ou seja, menor avaliação possível para um time. Caso contrário retorna a pontuação do time propriamente dita.

Finalmente, foram escolhidos os atletas que irão compor o banco de reserva e o capitão do time. Para cada posição, exceto o técnico, os participantes do Cartola FC têm direito a um atleta para colocar no banco de reserva, com a restrição de custar menos que os atletas da posição do time titular. Assim, foram removidos dos dados de atletas aqueles que já compõem o time titular e os que são mais caros do que o atleta de menor valor para cada posição. Posteriormente, para cada posição, será escolhido o atleta com maior potencial de pontuação para compor o banco de reserva, sendo que, caso não tenha nenhum atleta disponível, o banco de reserva daquela posição será ignorado. Destaca-se que, caso tenha escolhido um time sem o lateral, isto é, com esquema 3-4-3 ou 3-5-2, não será possível escolher um lateral para o banco de reserva. Já a definição do capitão é a escolha do atleta que tem o maior potencial de pontuação.

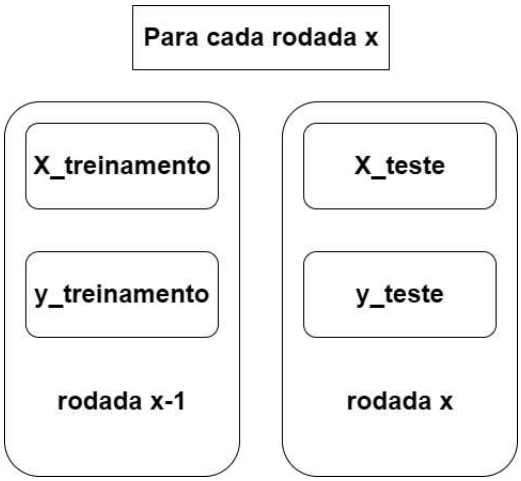
No final do processo, foram escolhidos 7 times, com 12 atletas cada, divididos por posições, conforme o esquema tático; um capitão entre os 11 atletas de campo e, no máximo, 5 atletas para o banco de reservas.

4 Resultados e Discussão

O objetivo deste trabalho é criar modelos para escalar predição de escalação de um time de Cartola FC cada rodada, considerando os esquemas táticos e o limite financeiro, a partir da obtenção dos dados, limpeza, avaliação dos potenciais de pontuação dos atletas usando algoritmos de aprendizado de máquina, montagem e avaliação do time via algoritmo genético, redução de variáveis com métodos de importância de atributos e, novamente, estrutura o time para serem reavaliados.

Para cada rodada do segundo turno do campeonato brasileiro de futebol, isto é, desde a rodada 20 até a 38, foram aplicados os métodos para a escalação e avaliação dos times. Os dados de treinamento correspondem a todas as rodadas anteriores à rodada de predição. Já os dados de teste são os dados relacionados à rodada de predição, sem a pontuação dos atletas, pois esta informação equivale ao objetivo do presente trabalho. A figura 2 representa um diagrama da divisão dos dados. Esse mecanismo visa simular o processo de escalar um time na vida real.

Figura 2 – Diagrama de treinamento e teste de uma rodada x.



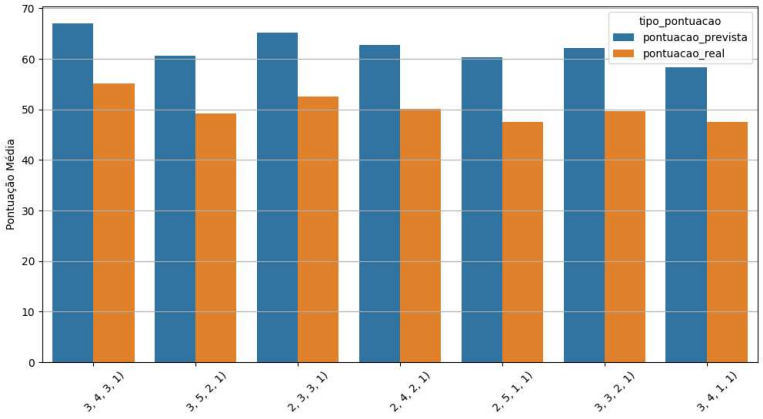
Para os resultados apresentados neste capítulo, as pontuações consideram o bônus de capitão e o banco de reserva.

4.1 Resultados por Formação Tática

Nesta seção são apresentados os resultados dos testes por formação tática, comparando as pontuações médias e máximas das pontuações previstas e reais, juntamente com os respectivos desvios padrão, sem considerar outras variantes que serão apresentados em sequência.

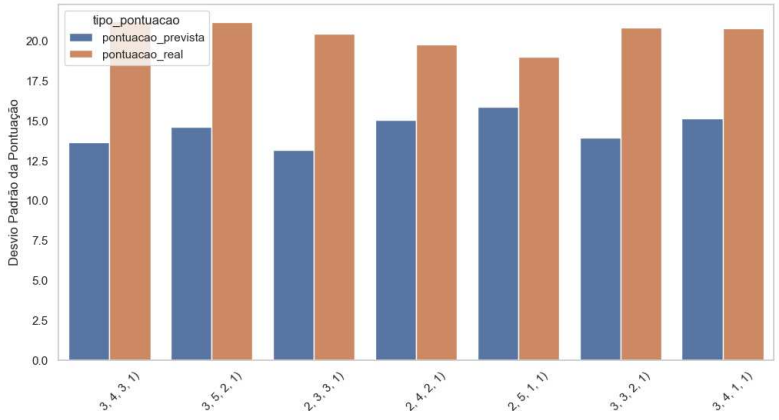
Com relação a média de pontuações, todos os esquemas tiveram resultados parecidos, com leve vantagem para formações de três zagueiros, quatro meias e três atacantes (conhecido como 3-4-3 e dois laterais, dois zagueiros, três meias e três atacantes (conhecido como 4-3-3), apresentado na figura 3.

Figura 3 – Pontuação média por Formação Tática



Com relação ao desvio padrão das pontuações, todos os esquemas táticos tiveram como esperado em torno de 12.5 à 15 pontos de desvio. Contudo, na prática, o desvio padrão foi maior, com pontuações acima de 18 pontos, apresentado na figura 4.

Figura 4 – Desvio padrão das pontuações por Formação Tática



Em relação à média das pontuações máximas por rodada, isto é, a média de todas as melhores pontuações alcançadas por rodada para cada formação tática, o resultado foi semelhante, e as formações 3-4-3 e 4-3-3 se saíram à frente das demais, como indica na figura 5.

Em alguns casos, o desvio padrão das pontuações máximas por rodada foram de acordo com o previsto. Contudo, na maioria das vezes, o desvio esperado foi inferior em relação ao desvio previsto, indicado na figura 6.

Figura 5 – Média da pontuações máximas por rodada

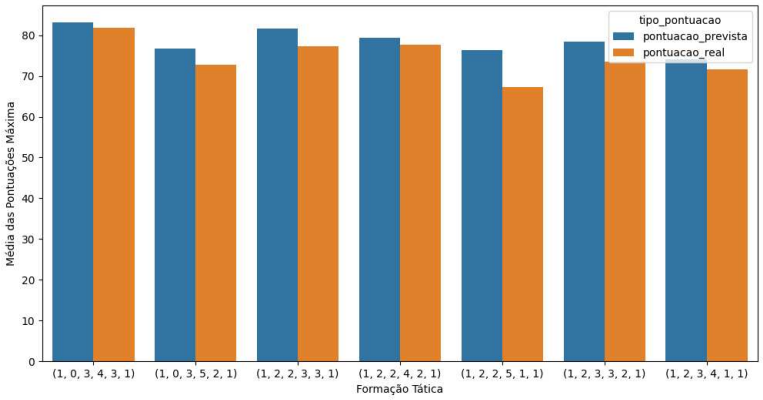
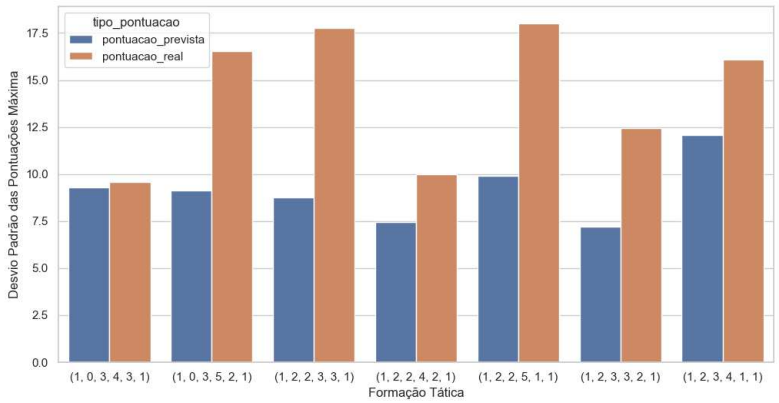


Figura 6 – Desvio padrão das pontuações máximas por rodada



4.2 Resultados por Modelo de Predição

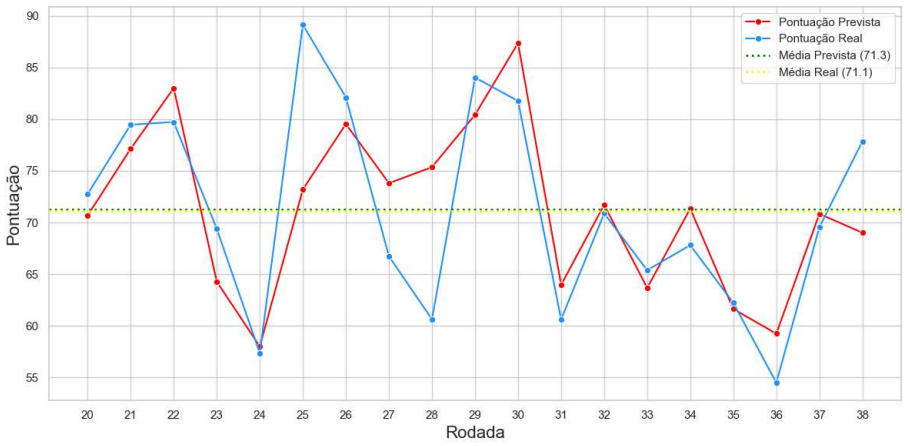
Nesta seção são apresentados os resultados dos testes por modelo, juntamente a comparação entre a pontuação prevista, a pontuação real e a diferença entre essas pontuações, estabelecendo limites de cartoletas. É considerado o modelo que obteve a melhor pontuação por rodada, a partir de fatores como cartoletas e variáveis do algoritmo genético (número de gerações, tamanho da população e taxa de mutação).

4.2.1 Resultados do Catboost

O Algoritmo de Catboost apresentou bons resultados ao longo do campeonato. No geral, a diferença entre a pontuação prevista e a pontuação real é pequena. Na figura 7, a média da pontuação prevista ao longo do segundo turno é de 71.3 pontos, enquanto a média real é de 71.1 pontos, quando o número de cartoletas foi estabelecido para 80.

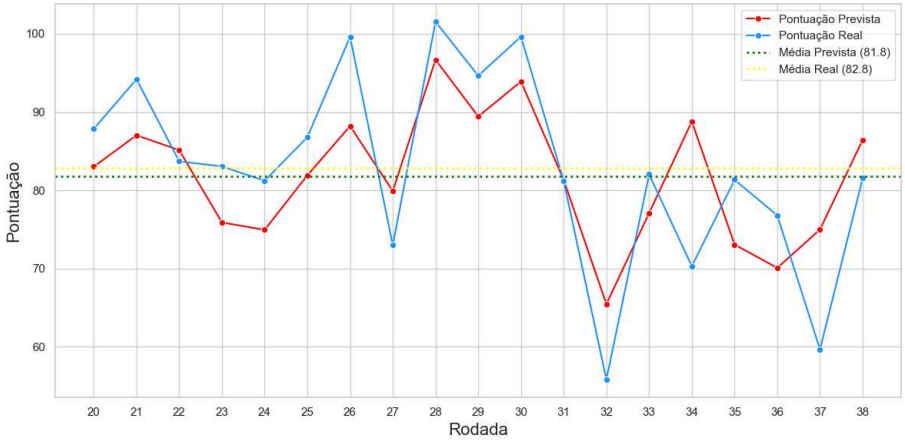
Já quando as cartoletas foram fixadas em 100, as médias aumentaram, com a média da pontuação prevista de 81.0 pontos e média da pontuação real de 82.8 pontos,

Figura 7 – Pontuação Prevista x Pontuação Real do Catboost com Cartoleta = 80



apresentado na figura 8.

Figura 8 – Pontuação Prevista x Pontuação Real do Catboost com Cartoleta = 100



A diferença entre a pontuação prevista e pontuação real pode ser vista nas figuras a seguir. Para número de cartoletas igual a 80, apresentado na figura 9 o erro médio variou no intervalo de $[-5.74, 4.49]$, com a média de -0.8. Já o maior erro registrado positivamente, isto é, quando a pontuação prevista é menor que a pontuação real, foi de 15.94, ocorrido na rodada 25, enquanto o maior erro negativo, isto é, a pontuação prevista é maior que a pontuação real, foi de -18.97, na rodada 28.

Já quando configuramos as cartoletas para 100, identificado na figura 10 a média dos erros ficou no intervalo de $[-11.24, 6.63]$ e a sua média foi um pouco maior, 0.1. A maior disparidade positiva foi de 17.14, registrado na rodada 33; enquanto a maior disparidade negativa foi de -18.50, na rodada 23.

Figura 9 – Diferença de Pontuação do Catboost com Cartoleta = 80

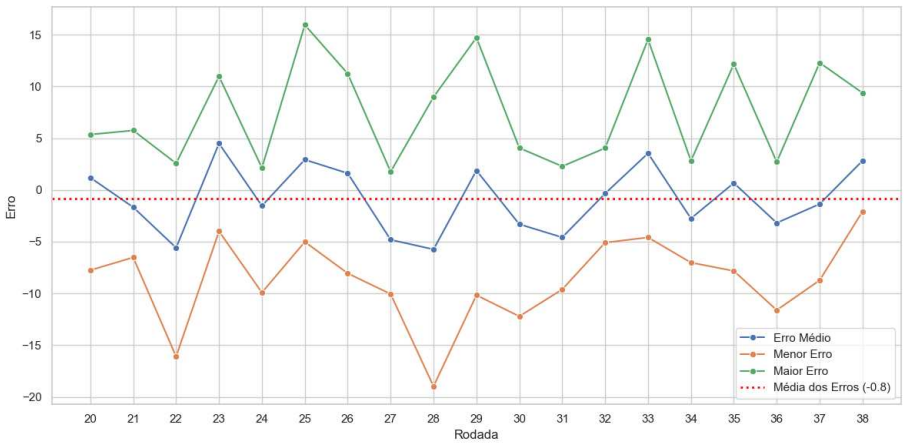
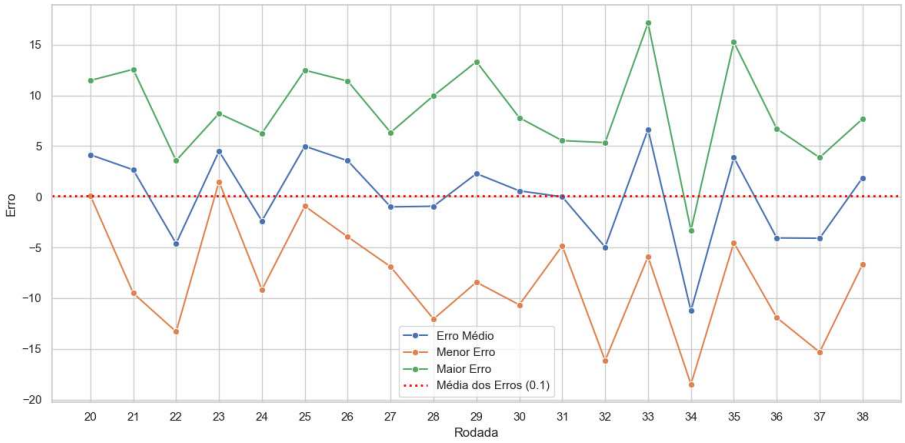


Figura 10 – Diferença de Pontuação do Catboost com Cartoleta = 100



4.2.2 Resultados da Floresta Aleatória

O algoritmo de Floresta Aleatória também apresentou pontuações consideradas boas, sendo levemente inferior comparado ao Catboost. A figura 11 mostra que a média das pontuações prevista com 80 cartoletas foi de 70.0 pontos, enquanto a média real foi de 69.9.

Já a tabela 12 mostra os resultados de quando o número de cartoletas foi de 100, a média prevista passou para 78.7 e a média real, para 79.9.

Na tabela 13, é informado que a média dos erros com cartoletas igual a 80 variou no intervalo $[-11.46, 5.25]$. Já o maior erro positivo foi de 16.03, registrado na rodada 29, enquanto o maior erro negativo foi de -16.83, na rodada 23.

Com 100 cartoletas, a média dos erros ficou entre $[-8.02, 15.35]$. Já o maior erro positivo foi de 32.79, registrado na rodada 29, enquanto o maior erro negativo foi de -20.29, na rodada 33, como visto na figura 14

Figura 11 – Pontuação Prevista x Pontuação Real do RF com Cartoleta = 80

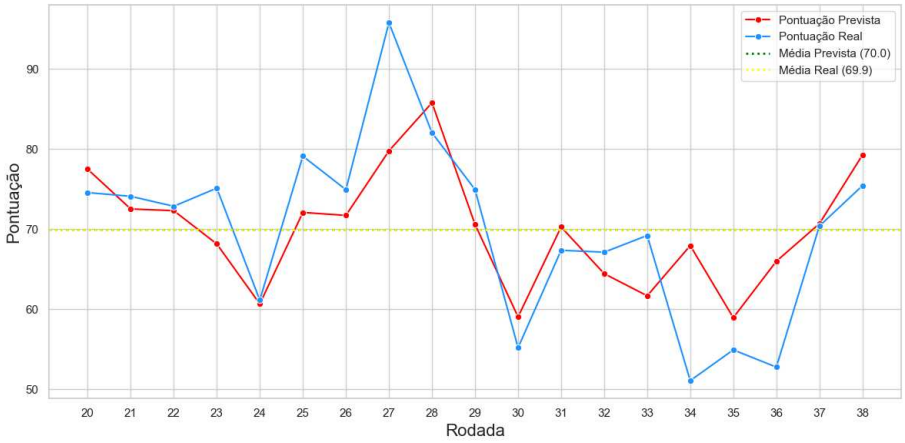


Figura 12 – Pontuação Prevista x Pontuação Real do RF com Cartoleta = 100

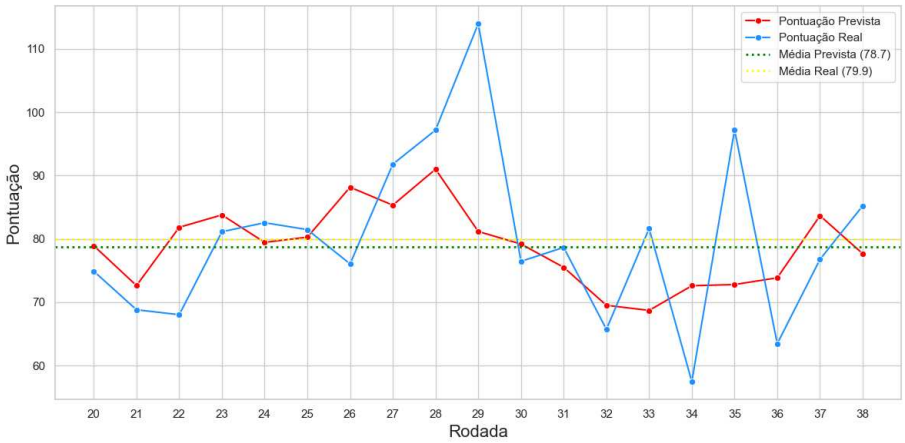


Figura 13 – Diferença de Pontuação do RF com Cartoleta = 100

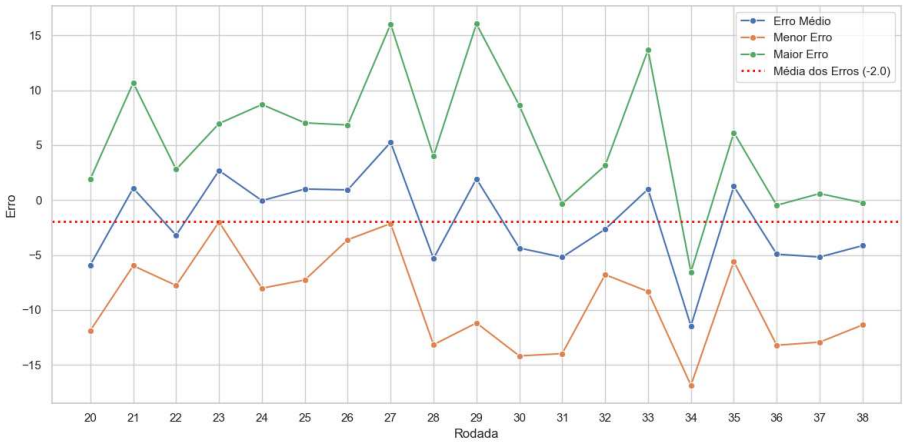
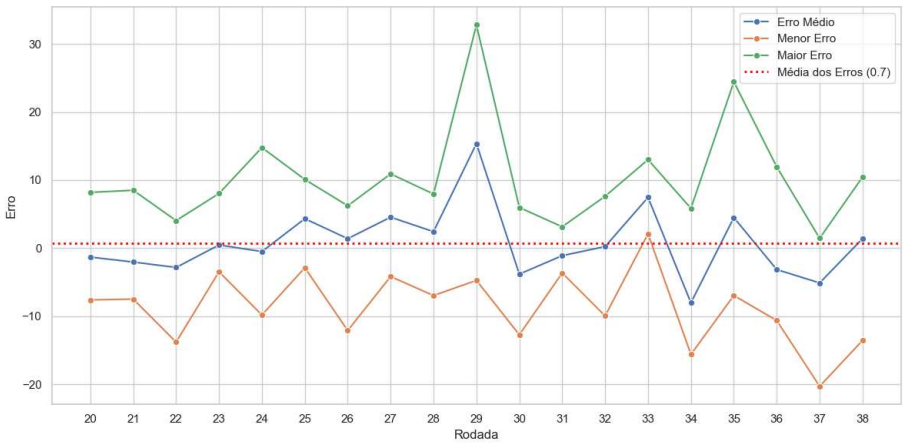


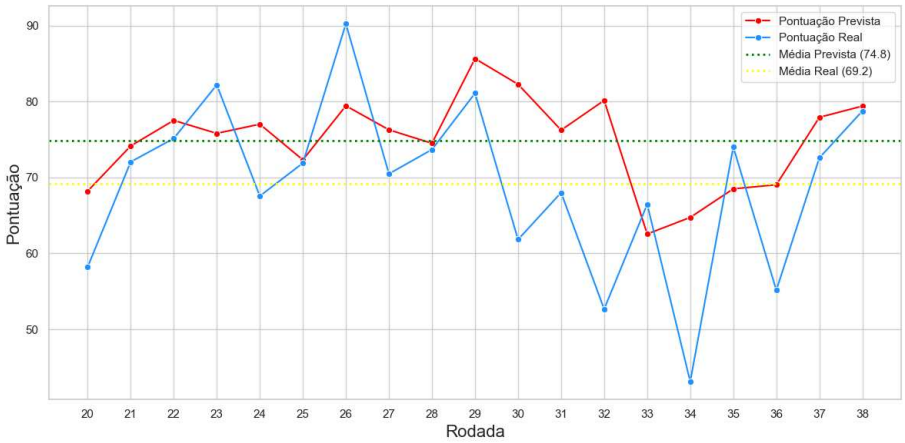
Figura 14 – Diferença de Pontuação do Catboost com Cartoleta = 100



4.2.3 Resultados de Máquina de Vetores de Suporte

A pontuação obtida pelo SVM também foi consideravelmente boa, sendo que a média prevista foi de 74.8 pontos e a média real, 69.2 pontos, para 80 cartoletas, indicado na tabela 15 e a média prevista igual a 83.1 pontos e a média real, 79.4 pontos para 100 cartoletas, indicado na figura 16. Contudo, a diferença entre a média prevista e a média geral foi levemente maior comparado aos dois algoritmos citados anteriormente.

Figura 15 – Pontuação Prevista x Pontuação Real do SVM com Cartoleta = 80



Na figura 17, a média dos erros com cartoletas igual 80 mostra que variou entre $[-16.33, 2.62]$. Já o maior erro positivo foi de 14.29, registrado na rodada 38, enquanto o maior erro negativo foi de -27.47, na rodada 26.

Na figura 18, a média dos erros com cartoletas = 100 variou entre [-15.35, 3.72]. Já o maior erro positivo foi de 16.55, registrado na rodada 35, enquanto o maior erro negativo foi de -23.79, na rodada 29.

Figura 16 – Pontuação Prevista x Pontuação Real do SVM com Cartoleta = 100

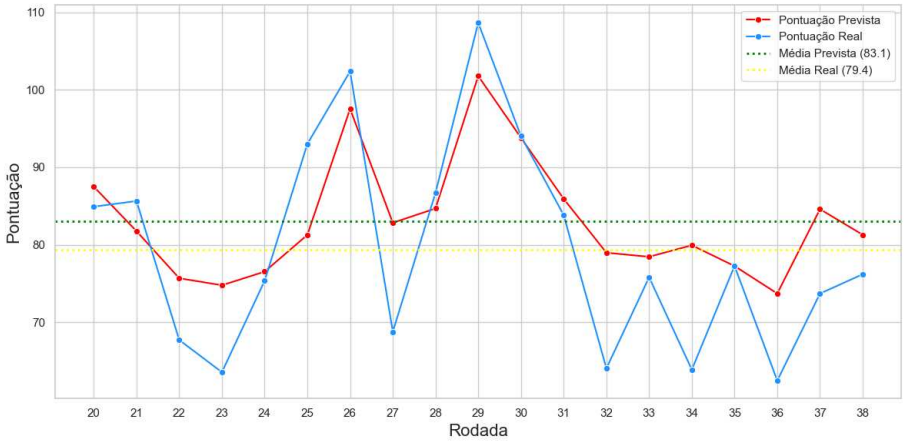


Figura 17 – Diferença de Pontuação do SVM com Cartoleta = 80

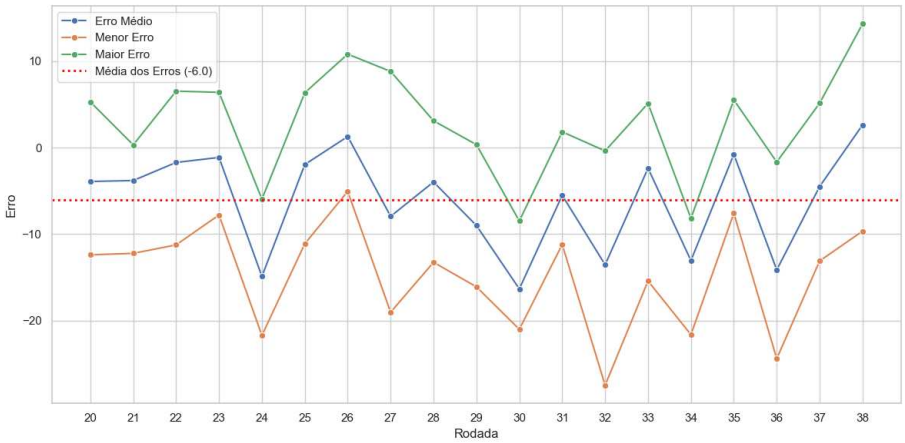
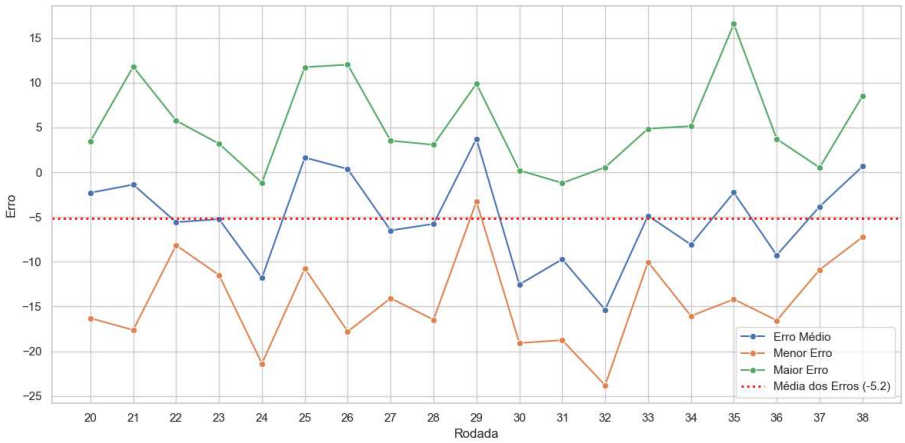


Figura 18 – Diferença de Pontuação do SVM com Cartoleta = 100



4.2.4 Resultados da Rede Neural Recorrente

A RNN teve resultado inferior comparado aos outros três algoritmos. Para cartoletas = 80, a média prevista foi de 66.6, com a média real sendo 53.3, enquanto, para cartoletas = 100, a média prevista foi de 75.1, com a média real sendo 61.1, como mostram as figuras 19 e 21, respectivamente.

Figura 19 – Pontuação Prevista x Pontuação Real do RNN com Cartoleta = 80

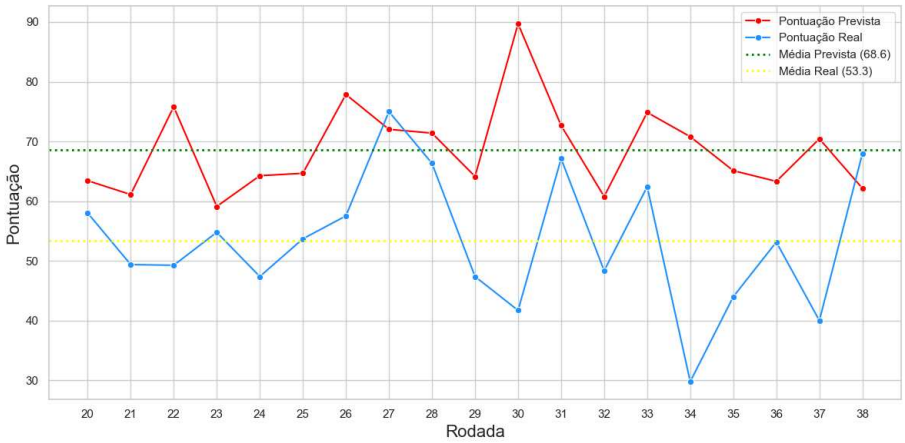
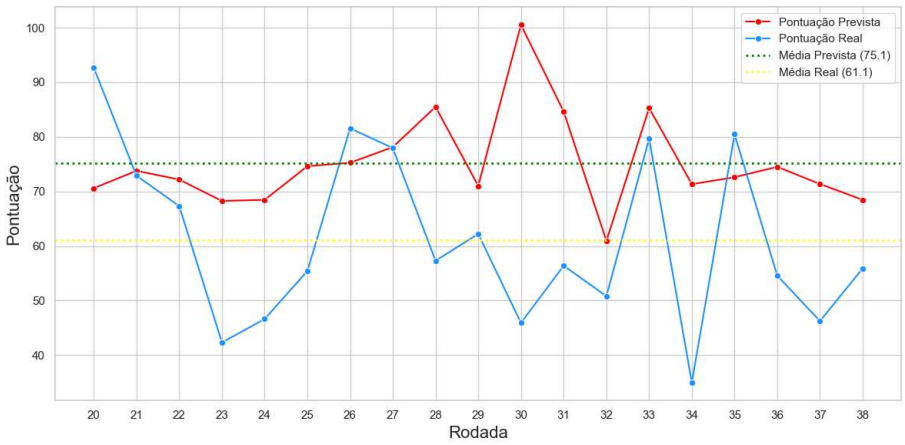


Figura 20 – Pontuação Prevista x Pontuação Real do RNN com Cartoleta = 100



Na figura 21, identifica-se que a média dos erros com cartoletas = 80 variou entre $[-22.13, 2.37]$. Já o maior erro positivo foi de 27.00, registrado na rodada 21, enquanto o maior erro negativo foi de -47.99, na rodada 23.

Na figura 22, é possível ver que a média dos erros com 100 cartoletas variou entre $[-21.20, 7.44]$. Já o maior erro positivo foi de 38.23, registrado na rodada 25, enquanto o maior erro negativo foi de -54.63, na rodada 21.

Figura 21 – Diferença de Pontuação do RNN com Cartoleta = 80

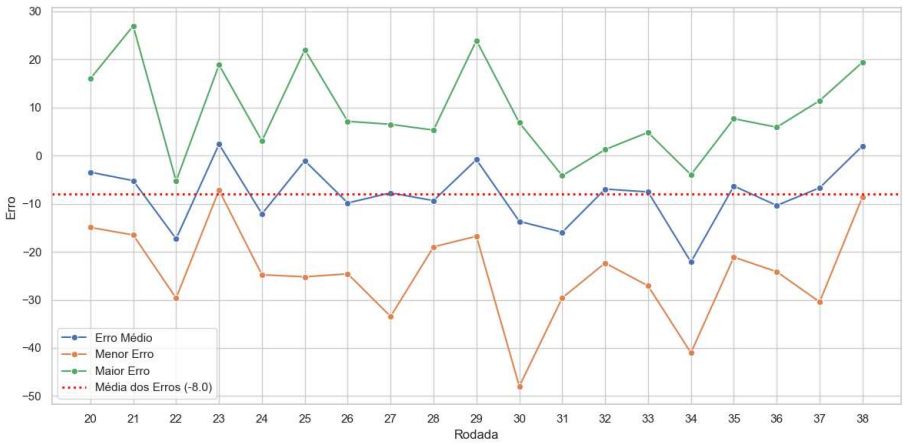
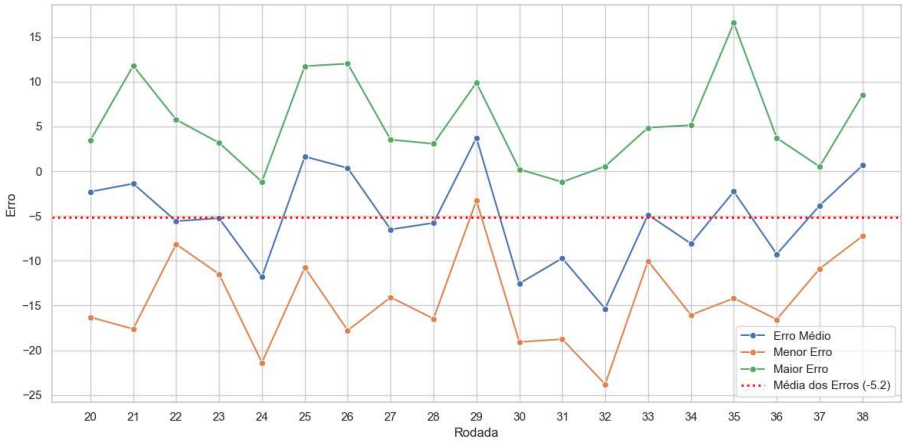


Figura 22 – Diferença de Pontuação do RNN com Cartoleta = 100



4.3 Resultados por Limite de Cartoletas

Nesta seção são apresentados os resultados conforme o limite de cartoletas. As figuras 23 e 24 significam uma média das pontuações e a comparação entre as pontuações previstas e reais dos modelos apresentadas acima, separados por cartoletas.

No geral, o resultado apresentado com limite de cartoletas igual a 80 foi boa. Contudo, algumas disparidades são notadas, como na rodada 30, com diferença de 47.99 pontos, e na rodada 32, com diferença de 27.42 pontos.

A diferença nas médias também foram considerável. A média prevista foi de 78.00 pontos, enquanto a média real foi de 74.00 pontos, resultado em 4.00 pontos de diferença.

O desempenho dos algoritmos com limite de 100 cartoletas também foi boa, conforme as figuras 25 e 26. No entanto, também apresentou disparidades, como nas rodadas 30 e 36, com 54.63 e 19.94 pontos de diferença, respectivamente.

Em relação a diferença das médias, a discrepância foi ainda maior, com 86.8 pontos de média prevista e 81.5 pontos de média real, resultando em 5.3 pontos de diferença entre

Figura 23 – Média por Rodada com Cartoleta = 80

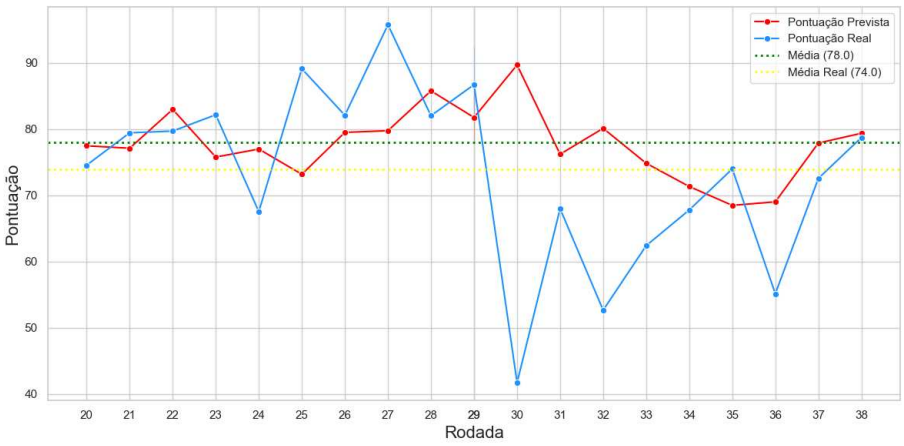
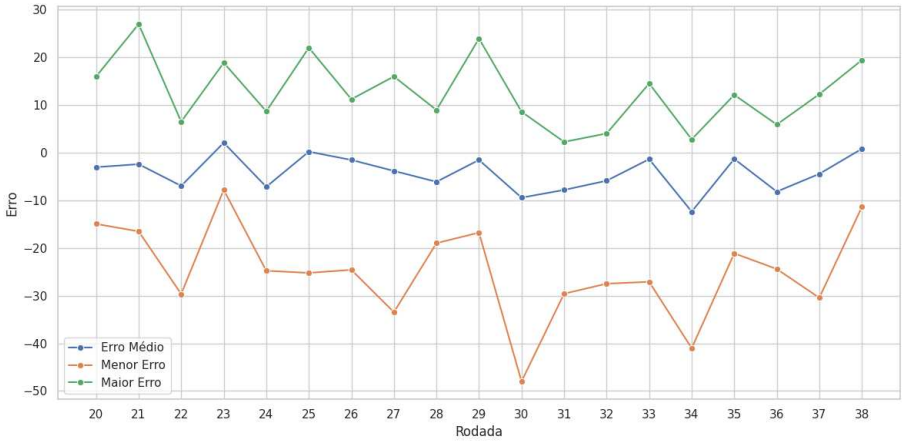


Figura 24 – Pontuação Prevista x Pontuação Real por Rodada com Cartoleta = 80



as médias.

Figura 25 – Pontuação Prevista x Pontuação Real por Rodada com Cartoleta = 100

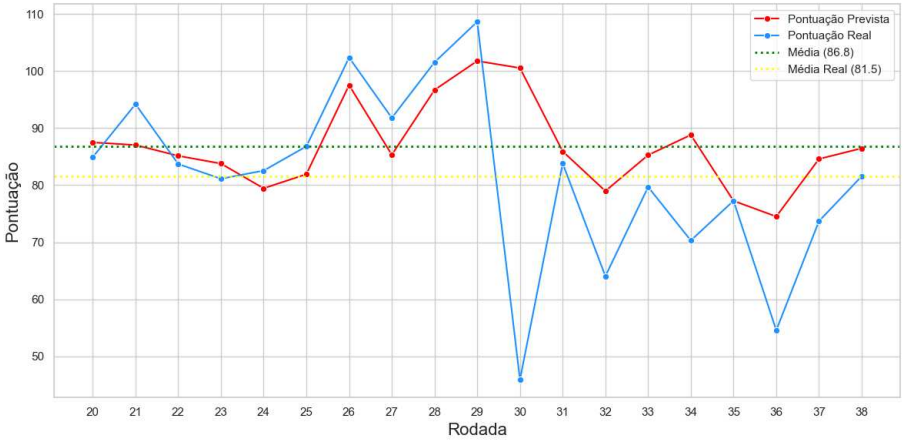
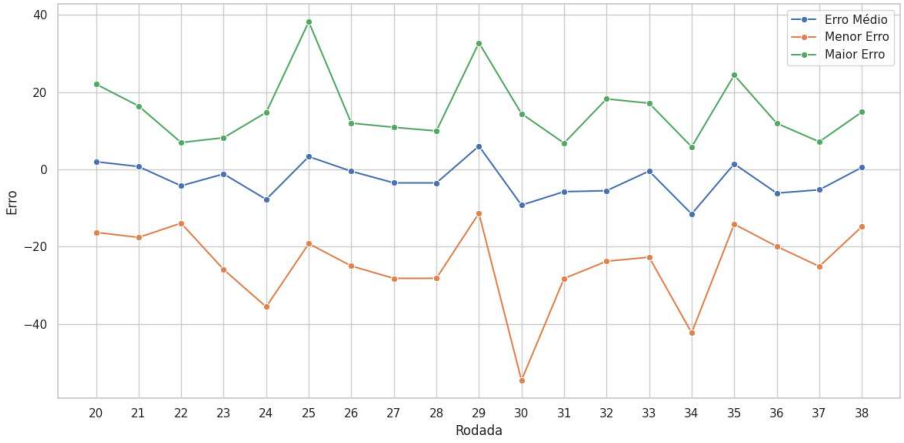


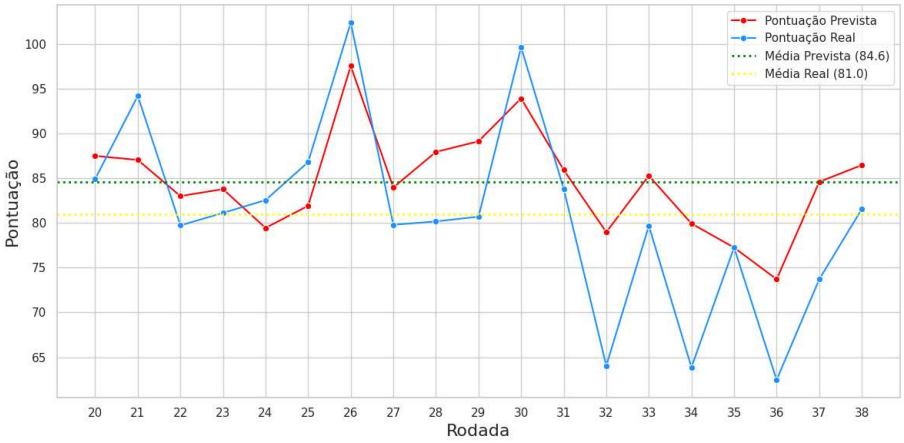
Figura 26 – Pontuação Prevista x Pontuação Real por Rodada com Cartoleta = 100



4.4 Resultados por Número de Gerações

Nesta seção será apresentado os resultados conforme o número de gerações do algoritmo genético. Quanto ao número de gerações igual 50, o resultado foi bom, com a média prevista de 84.6 pontos e média real de 81.0 pontos, apresentado nas figuras 27 e 28. Contudo, nota-se que também tiveram algumas disparidades entre as médias, como nas rodadas 32, 34 e 37, com 14.93, 16.05 e 10.89 pontos de diferença, respectivamente.

Figura 27 – Média das pontuações por Rodada com número de gerações = 50



O resultado apresentado com o número de gerações igual a 100 foi inferior em comparação com número de gerações igual a 50, mesmo mantendo a média boa, como consta nas figuras 27 e 30. A média prevista foi 82.6 pontos, enquanto a média real foi 77.2 pontos. Contudo, destaca-se uma disparidade muito grande nas rodadas 30, 32 e 36, com 54.63, 27.47 e 19.94 pontos de diferença entre eles.

Figura 28 – Pontuação Prevista x Pontuação Real por Rodada com número de gerações = 50

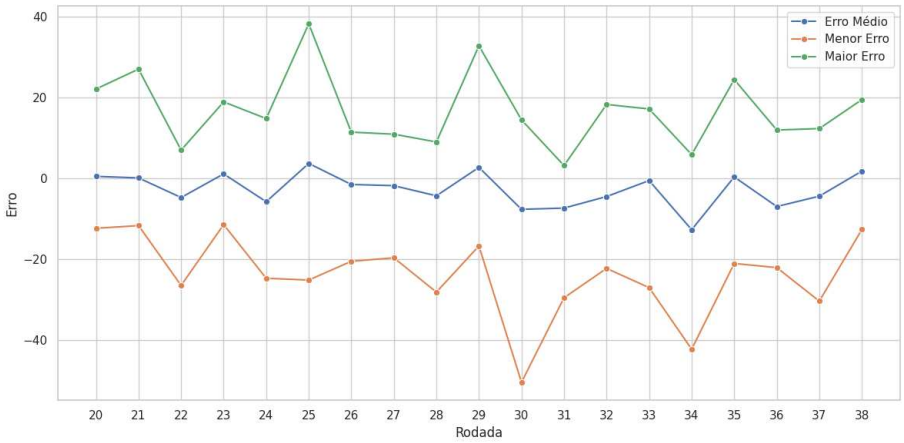


Figura 29 – Média das pontuações por Rodada com número de gerações = 100

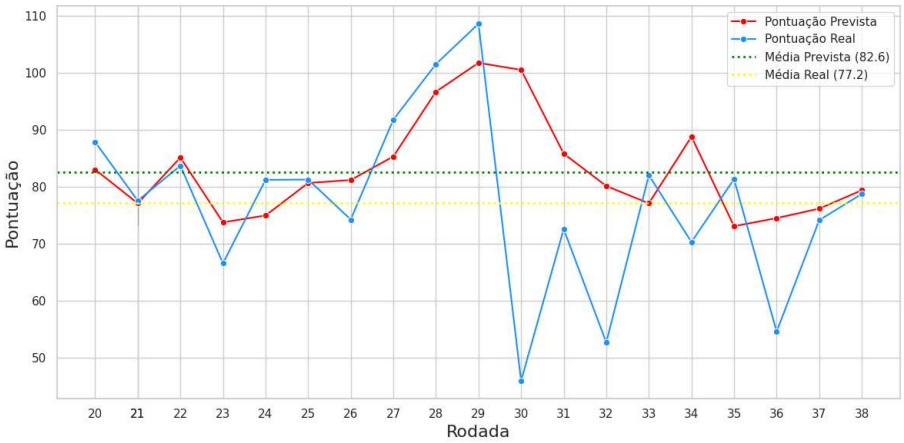
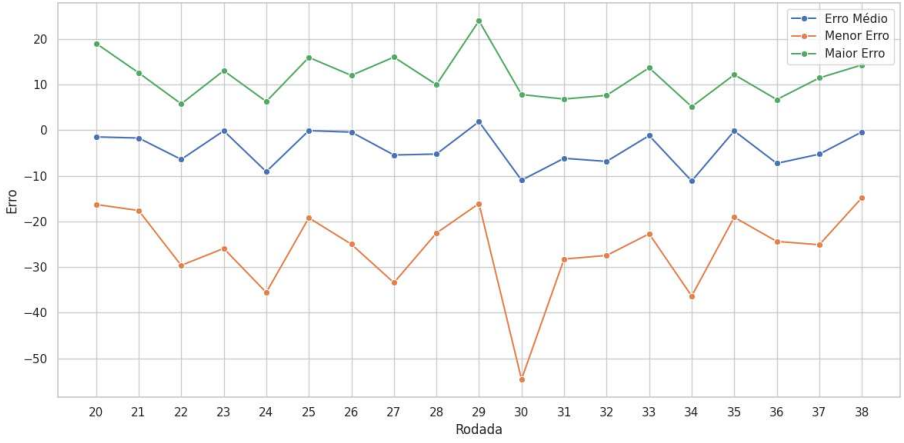


Figura 30 – Pontuação Prevista x Pontuação Real por Rodada com número de gerações = 100



4.5 Resultados por Tamanho da População

Nesta seção são apresentados os resultados conforme o tamanho da população do algoritmo genético. Com uma população de 20 indivíduos, o resultado foi bom, com a média prevista de 83.4 pontos e média real de 78.9 pontos. Também houve disparidades entre as médias, como nas rodada 30 com 50.53 pontos de diferença. Os resultados estão apresentados nas figuras 31 e 32.

Figura 31 – Média das pontuações por Rodada com tamanho da população = 20

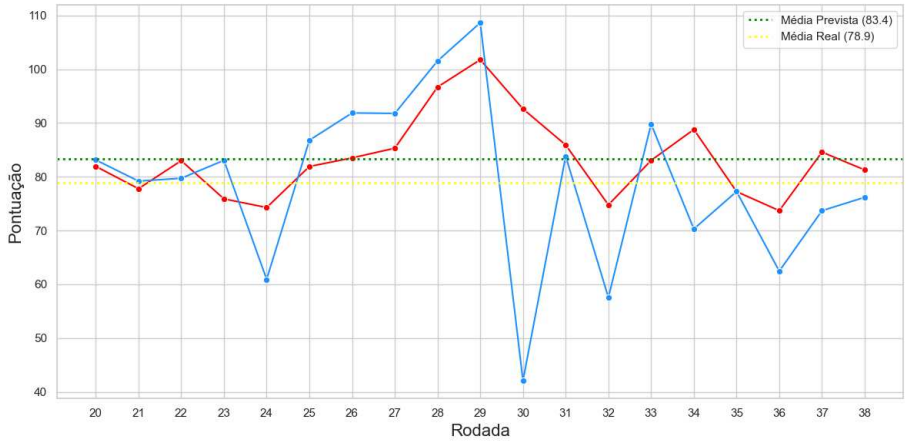
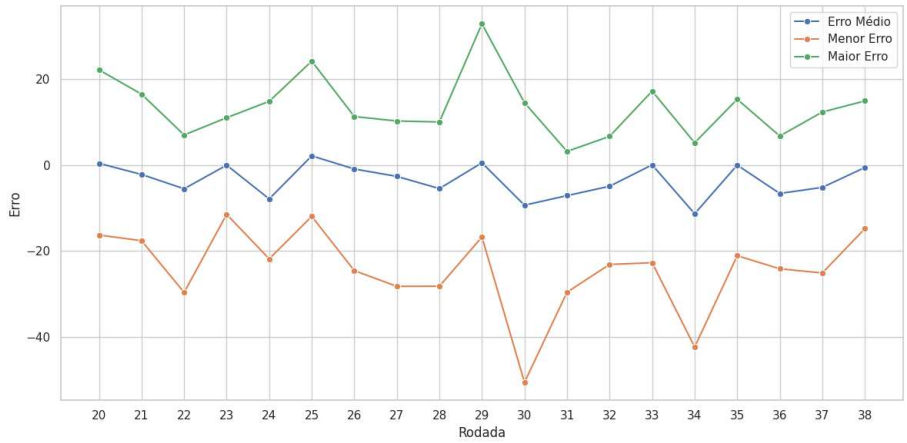


Figura 32 – Pontuação Prevista x Pontuação Real por Rodada com tamanho da população = 20



Já nas figuras 33 e 34, os resultados apresentados com o tamanho da população igual a 30 foram razoáveis, mesmo mantendo a média boa. A média prevista foi 85.2 pontos, enquanto a média real foi 79.8 pontos. Contudo, destaca-se uma disparidade muito grande nas rodadas 30, 35 e 36, com 54.63, 24.47 e 19.94 pontos de diferença entre eles. No entanto, diferente das outras demais que foram apresentadas, a diferença de pontuações da rodada 35 foi positiva.

Figura 33 – Média das pontuações por Rodada com tamanho da população = 30

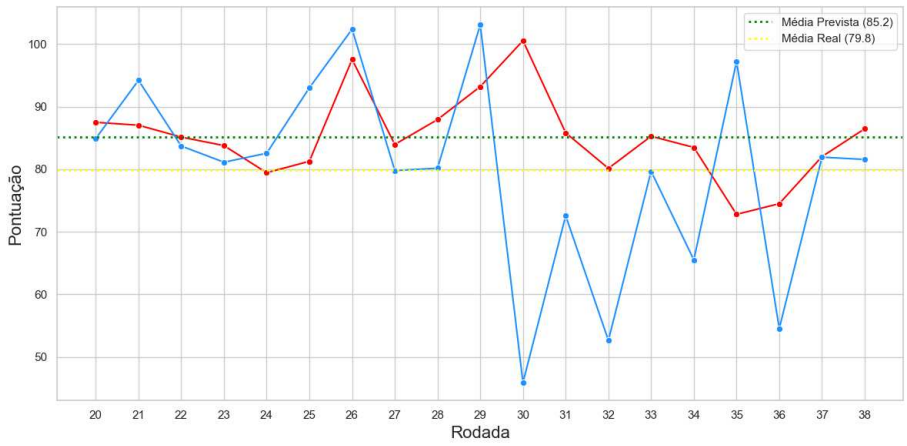
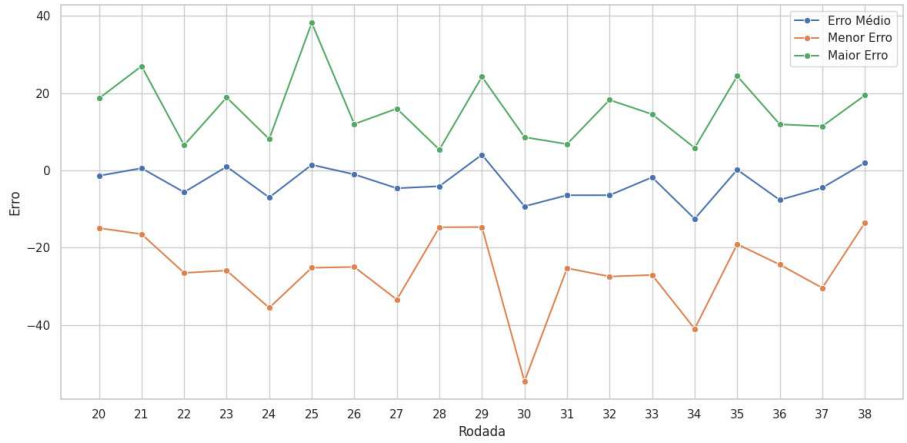


Figura 34 – Pontuação Prevista x Pontuação Real por Rodada com tamanho da população = 30



4.6 Resultados por Taxa de Mutação

Nesta seção será apresentado os resultados conforme a taxa de mutação do algoritmo genético.

Nas figuras 36 e 35, foram apresentados resultados com taxa de mutação de 0.01, nos quais foram considerados bons, com a média prevista de 83.4 pontos e média real de 78.9 pontos. Também tiveram disparidades entre as médias, como nas rodada 30 com 50.53 pontos de diferença.

Já com taxa de mutação igual a 0.05, representado pelas figuras 38 e 37, os resultados apresentados foram razoáveis, mesmo mantendo a média boa. A média prevista foi 85.2 pontos, enquanto a média real foi 79.8 pontos. Contudo, destaca-se uma disparidade muito grande nas rodadas 30, 35 e 36, com 54.63, 24.47 e 19.94 pontos de diferença entre eles. No entanto, diferente das outras demais que foram apresentadas, a diferença de pontuações da rodada 35 foi positiva.

Figura 35 – Média das pontuações por Rodada com taxa de mutação = 0.01

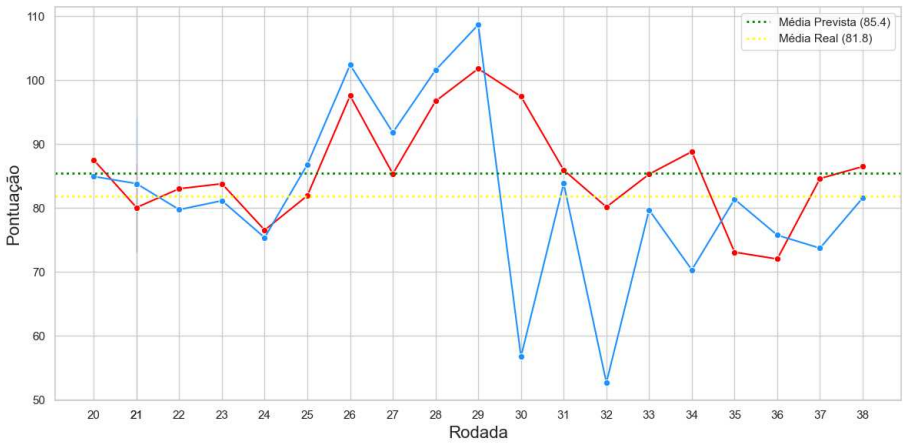


Figura 36 – Pontuação Prevista x Pontuação Real por Rodada com taxa de mutação = 0.01

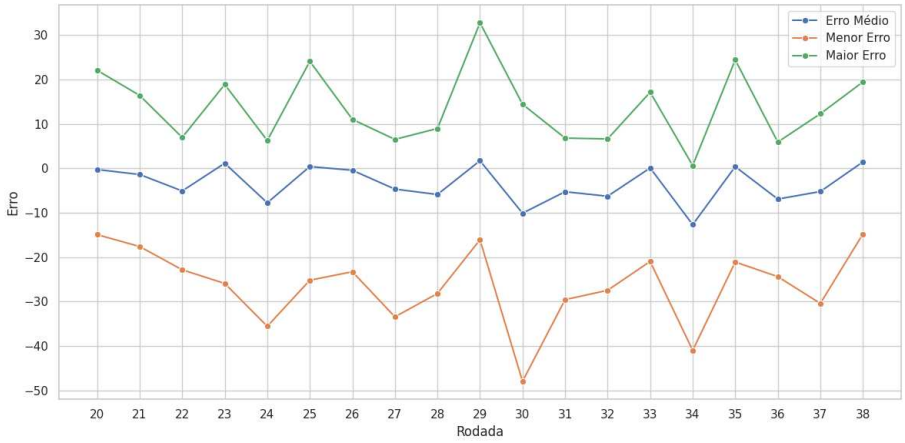


Figura 37 – Média das pontuações por Rodada com taxa de mutação = 0.05

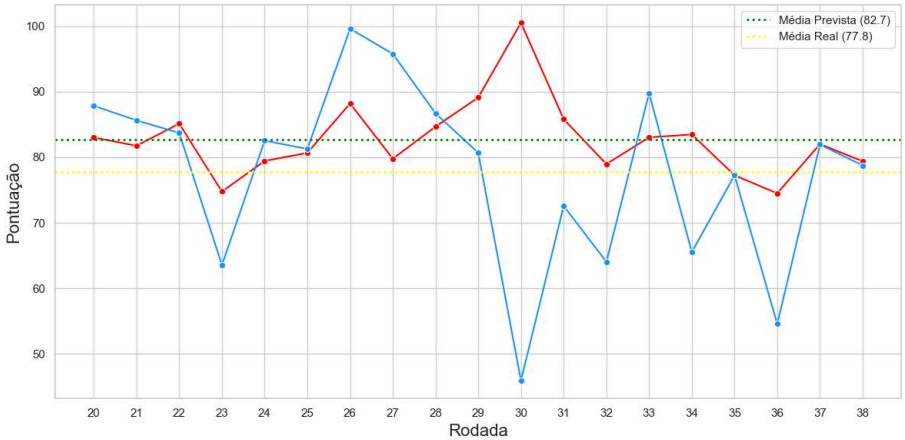
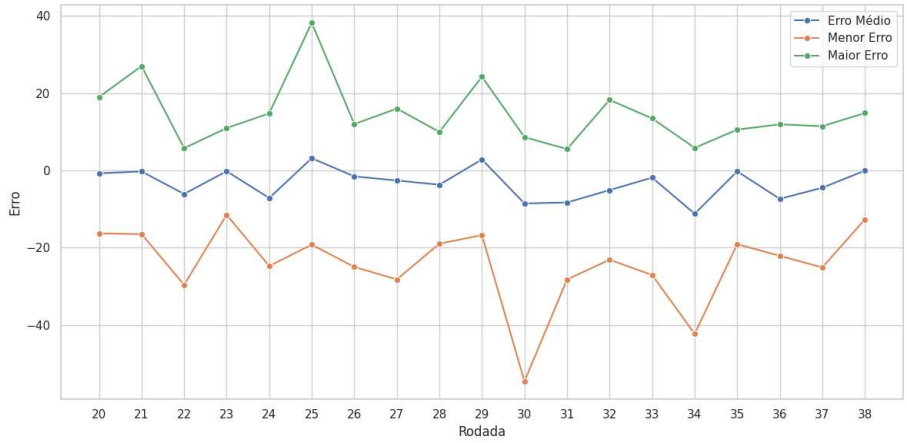


Figura 38 – Pontuação Prevista x Pontuação Real por Rodada com taxa de mutação = 0.05



4.7 Resultados por Medidas Estatísticas

Foram utilizados dois métodos para calcular o desempenho dos modelos criados neste trabalho, que foram apresentados nas figuras 39 e 40. Ambos os resultados foram razoáveis, com a média do erro médio quadrático (MSE) por rodada = 12.0 e a média do erro médio absoluto (MAE) por rodada = 2.5.

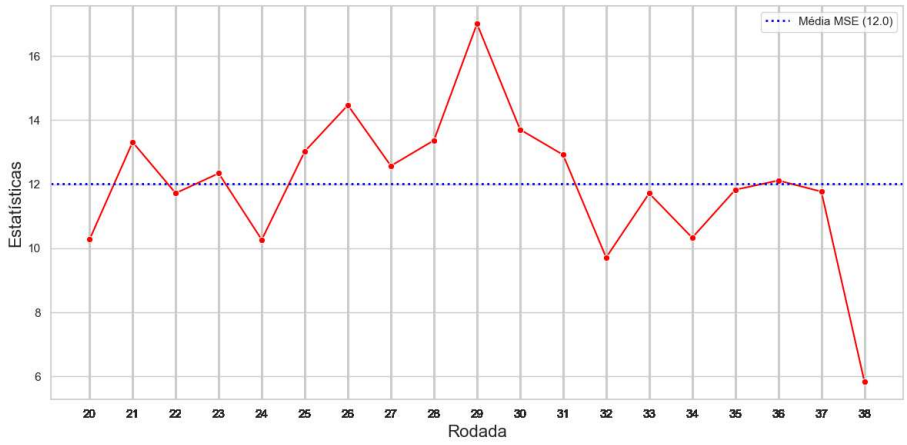


Figura 39 – Média do Erro Quadrático Médio por Rodada

4.8 Discussão

O melhor esquema tático, conforme os resultados apresentados neste capítulo, foi o esquema 3-4-3, ou seja, sem o uso de laterais, sendo o esquema 4-3-3 o segundo melhor. Contudo, esse resultado é interessante, visto que o esquema 4-3-3 é o mais escolhido entre as opções disponíveis, como informado pelo site do Cartola FC (GLOBO, 2024). Além disso, o esquema 3-4-3 teve o menor desvio padrão com relação a pontuações máximas por rodada, ao considerar o desvio real. Entretanto, a superioridade desses esquemas

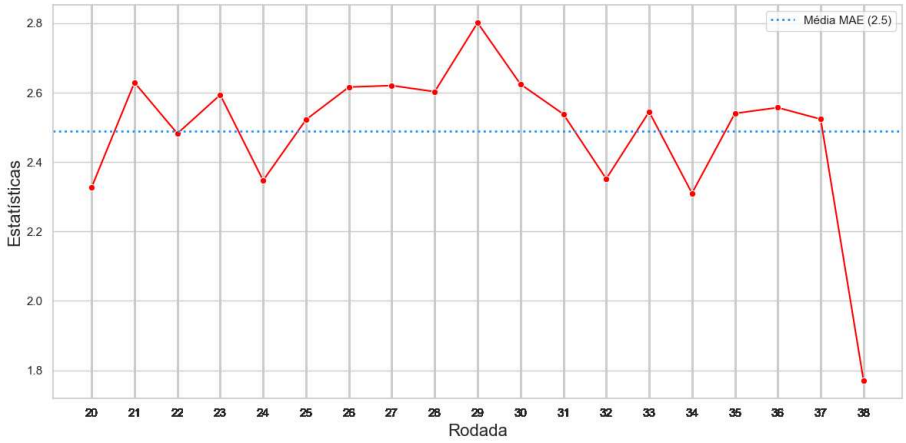


Figura 40 – Média do Erro Médio Absoluto por Rodada

em relação aos demais podem ser justificados pelo uso de mais atacantes em relação aos demais, justamente a posição onde é mais provável que um atleta possa alcançar pontuações mais altas.

No geral, os modelos criados apresentaram resultados satisfatórios ao longo do período previsto, mantendo uma boa média de pontuação, com exceção dos times baseados em Rede Neural Recorrente, que tiveram resultados abaixo comparado aos demais algoritmos. Além disso, em geral, o desempenho dos times com investimento maior apresentaram melhores resultados em comparação com os demais times, o que é um fator esperado. Contudo, as diferenças nos parâmetros das variáveis ligadas ao algoritmo genético, isto é, número de gerações, tamanho da população e taxa de mutação sugerem pouca mudança no resultado final em comparação com os demais fatores.

A respeito da comparação das pontuações previstas e reais, não houveram muitas disparidades, mesmo que, em algumas rodadas específicas, possam encontrar tais diferenças de forma significativa. Contudo, esse fator pode ser justificado pelo desempenho incomum dos atletas e do time numa determinada partida, ocorrido positivamente ou negativamente. Além disso, em certos confrontos, alguns atletas podem ser preservados, jogando menos tempo que o normal. Isso faz com que a possibilidade desses atletas de pontuarem diminua, o que influencia diretamente no desempenho do time escalado. Com relação às medidas estatísticas para justificar o desempenho dos modelos, podem ser considerados razoáveis, conforme mostrado na Seção 4.7.

5 Conclusão

Este trabalho propôs métodos para criar times no jogo Cartola FC de forma automática, com o intuito de atingir o máximo de pontos, estabelecendo limites de cartoletas. Para tanto, foram feitos uma sequência de tarefas, passando desde a coleta de dados, limpeza, criação do modelo a partir dos algoritmos de aprendizado de máquina, reunir as principais variáveis, refazer o modelo e criar o time, com base nos métodos do algoritmo genético.

Com relação aos dados, os repositórios apresentam informações satisfatórias, tanto em quantidade como em qualidade, mesmo com o uso de dados obtidos diretamente do site oficial de Cartola FC a partir de API, que, inclusive, funciona bem.

Ao analisar os resultados, é possível dizer que o trabalho foi satisfatório, mantendo uma boa média de pontuação ao longo do tempo estipulado. Contudo, alguns fatores interessantes foram notados.

Foi possível observar o desempenho de cada algoritmo de aprendizado de máquina que foram aplicados, juntamente com as mudanças nos valores das variáveis que fizeram parte do algoritmo genético. Uma outra contribuição deste trabalho foi atualizar alguns conceitos que passaram a existir desde os trabalhos passados, como o capitão e banco de reservas.

Para trabalhos futuros, é possível considerar aplicação de novas perspectivas. Por ter um calendário denso, muitas vezes os times brasileiros são submetidos a utilizar times alternativos em determinadas partidas, inclusive do campeonato brasileiro. Assim, dependendo da rodada, alguns times podem utilizar times mistos ou reservas, o que muda consideravelmente o seu desempenho e do adversário. Assim sendo, pode ser avaliado a utilização de dados em relação aos jogos dos outros campeonatos que os times vão enfrentar em curto prazo.

Além disso, a atualização anual do Cartola FC pode ser considerado um ponto de partida para novos projetos, como a mudança nas pontuações dos *scouts*, criação ou remoção dos *scouts* ou novas funcionalidades.

Como implicações prática do projeto, esse projeto pode ser introdutório para entender, de forma mais profunda, sobre o desempenho individual e coletiva de um time, destacado por *scouts*. Além disso, também é possível criar um método para prever os resultados de jogos e performance dos atletas para apostas esportivas.

Referências

AI catboost. **CatBoostRegressor**. 2025. Disponível em: <https://catboost.ai/docs/en/concepts/python-reference_catboostregressor>. Acesso em: 20 mar 2025. Citado na página 26.

AQUARELA ADVANCED ANALYTICS. 2017. Disponível em: <https://aquare.la/o-que-sao-outliers-e-como-trata-los-em-uma-analise-de-dados>. Acesso em: 20 mar 2025. Citado na página 24.

BREIMAN, L. Random forests. **Machine Learning**, Springer, v. 45, n. 1, p. 5–32, 2001. Citado 2 vezes nas páginas 16 e 17.

CATBOOST. **Feature Importance**. 2025. Disponível em: <https://catboost.ai/docs/en/concepts/fstr#fstr___regular-feature-importance>. Acesso em: 15 abr 2025. Citado na página 27.

COELHO FILHO, E. C. **Equações de Estimação Generalizadas na predição da pontuação de atacantes no Cartola FC**. 87 p. Monografia (Trabalho de Conclusão de Curso) — Universidade Federal de São Carlos, 2021. Disponível em: <https://repositorio.ufscar.br/handle/ufscar/13996>. Citado na página 20.

CORTES, C.; VAPNIK, V. Support-vector network. **Machine Learning**, Springer, v. 20, n. 3, p. 273–297, 1995. Citado na página 16.

CRUZ, D. L. da; SOUSA, J. V. M.; CALÇADA, D. B. Utilizando inteligência artificial explicável para formação de perfil de jogadores no Cartola FC. **CIS - Conjecturas Inter Studies**, v. 22, n. 12, p. 968–985, 2020. Disponível em: <https://doi.org/10.53660/CONJ-1627-2E61>. Citado na página 20.

DATASCIENCESPHERE. **Z-Score to identify and remove outliers | Exploratory Data Analysis**. 2024. Disponível em: <https://medium.com/@datasciencejourney100_83560/z-score-to-identify-and-remove-outliers-c17382a4a739>. Acesso em: 20 mar 2025. Citado na página 25.

DRUCKER, H.; BURGES, C. J. C.; KAUFMAN, L.; SMOLA, A.; VAPNIK, V. Support vector regression machine. **Advances in Neural Information Processing Systems**, v. 9, 1997. Citado na página 16.

DUQUE, A. **Campeonato Brasileiro de futebol**. 2024. Disponível em: <https://www.kaggle.com/datasets/adaoduque/campeonato-brasileiro-de-futebol>. Acesso em: 29 abr 2025. Citado na página 21.

EIBEN, A. E.; SMITH, J. E. **Introduction to Evolutionary Computing**. 2. ed. Alemanha: Springer-Verlag, 2015. 287 p. Citado na página 19.

FOOTHUB. **A inteligência artificial no futebol: O futuro dos gramados**. 2021. Disponível em: <https://foothub.com.br/a-inteligencia-artificial-no-futebol-o-futuro-dos-gramados>. Acesso em: 26 abr 2024. Citado na página 11.

GALA, A. S. **Aprendizado de Máquina: entenda o que é o Machine Learning!** 2023. Disponível em: <https://www.handtalk.me/br/blog/aprendizado-de-maquina/>. Acesso em: 15 mar 2024. Citado na página 15.

GLOBO. **Cartola FC**. 2024. Disponível em: <https://cartola.globo.com/>. Acesso em: 25 fev 2024. Citado 3 vezes nas páginas 11, 13 e 46.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. Reading, MA: Addison-Wesley, 1989. Citado na página 19.

GOMIDE, H. **caRtola**. 2023. Disponível em: <https://github.com/henriquepgomide/caRtola>. Acesso em: 25 fev 2024. Citado na página 21.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado na página 18.

IBM. **O que são máquinas de vetores de suporte (SVMs)?** 2023. Disponível em: <https://www.ibm.com/br-pt/think/topics/support-vector-machine>. Acesso em: 06 maio 2025. Citado na página 16.

JEREMIAH, O. **CatBoost in Machine Learning: A Detailed Guide**. 2024. Disponível em: <https://www.datacamp.com/tutorial/catboost>. Acesso em: 19 may 2025. Citado na página 26.

KERAS. **Keras: Recurrent Layers**. 2025. Disponível em: <https://keras.io/api/layers/recurrent_layers>. Acesso em: 20 mar 2025. Citado na página 26.

LIMA, M. **O que são variáveis independentes e dependentes?** 2022. Disponível em: <https://www.blog.psicometriaonline.com.br/o-que-sao-variaveis-independentes-e-dependentes/>. Acesso em: 20 mar 2025. Citado na página 25.

LOUPPE, G. **Understanding Random Forests: From Theory to Practice**. 211 p. Tese (Doutorado) — Université de Liège, Belgium, out. 2014. Citado na página 17.

MANIAUDET, G.; SILVA, L. Brasileirão 2023 tem a maior média de público da história da competição. **GE**, dez. 2023. Disponível em: <https://ge.globo.com/espiao-estatistico/noticia/2023/12/08/brasileirao-2023-tem-a-maior-media-de-publico-da-historia-da-competicao-veja-balanco.ghtml>. Acesso em: 25 fev 2024. Citado na página 11.

MITCHELL, T. M. **Machine Learning**. [S.l.]: McGraw-Hill, 1997. Citado na página 15.

MUELLER, J. P.; MASSARON, L. **Machine Learning for Dummies**. 2. ed. Hoboken, NJ: For Dummies, 2021. Citado 2 vezes nas páginas 11 e 15.

ORACLE. **RMSE (Root Mean Squared Error)**. 2025. Disponível em: <https://docs.oracle.com/en/cloud/saas/freeform/ffuuu/insights_metrics_RMSE.html>. Acesso em: 29 apr 2025. Citado na página 28.

PROKHORENKOVA, L.; GUSEV, G.; VOROBEV, A.; DOROGUSH, A. V.; GULIN, A. Catboost: Unbiased boosting with categorical features. **arXiv preprint arXiv:1810.11363**, 2018. Disponível em: <https://arxiv.org/abs/1810.11363>. Citado 2 vezes nas páginas 17 e 18.

QUAL o papel do erro quadrado nos algoritmos de aprendizado de máquina? 2024. Disponível em: <https://www.linkedin.com/advice/0/what-role-does-mean-squared-error-play-machine-53stf?lang=pt&originalSubdomain=pt>. Acesso em: 20 mar 2025. Citado na página 28.

RIBEIRO, L. E. da S. **Predição de escalações para o jogo Cartola FC utilizando aprendizado de máquina e otimização**. 88 p. Monografia (Trabalho de Conclusão de Curso) — Universidade Federal de Uberlândia, 2019. Disponível em: <https://repositorio.ufu.br/handle/123456789/26681>. Citado na página 19.

SCIKIT-LEARN. **4.2. Permutation Feature Importance**. 2025. Disponível em: <https://scikit-learn.org/stable/modules/permutation_importance.html>. Acesso em: 15 abr 2025. Citado na página 27.

_____. **RandomForestRegressor**. 2025. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>. Acesso em: 20 mar 2025. Citado 2 vezes nas páginas 17 e 26.

_____. **SVR**. 2025. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html>. Acesso em: 20 mar 2025. Citado na página 26.

SHIN, T. **Understanding Feature Importance in Machine Learning**. 2023. Disponível em: <https://builtin.com/data-science/feature-importance>. Acesso em: 15 abr 2025. Citado na página 27.

SMOLA, A. J.; SCHÖLKOPF, B. A tutorial on support vector regression. **Statistics and Computing**, Springer, v. 14, n. 3, p. 199–222, 2004. Citado na página 16.

TREMONTI, L. **Cartola FC tem sucesso de engajamento em 2023 e terá novidades em 2024**. 2023. Disponível em: <https://cartolafcmix.com/humor/cartola-fc-tem-sucesso-de-engajamento-em-2023-e-tera-novidades-em-2024/>. Acesso em: 25 fev 2024. Citado na página 11.

VAQUER, G. Campeonato Brasileiro 2023 cresce 16% em audiência na TV paga. **O Tempo**, dez. 2023. Disponível em: <https://www.otempo.com.br/sports/futebol/campeonato-brasileiro-2023-cresce-16-em-audiencia-na-tv-paga-1.3296887>. Acesso em: 25 fev 2024. Citado na página 11.

VISCONDI, G. F.; JUSTO, D.; GARCÍA, N. M. **Aplicação de aprendizado de máquina para otimização da escalação de time no jogo Cartola FC**. 7 p. Monografia (Relatório Técnico) — Universidade de São Paulo, 2017. Disponível em: <https://www.researchgate.net/publication/321899230_Aplicacao_de_aprendizado_de_maquina_para_otimizacao_da_escalacao_de_time_no_jogo_Cartola_FC>. Citado na página 20.