HENRIQUE F. LOPES

APRENDIZADO DE MÁQUINA APLICADO A PREVISÃO DE DESEMPENHO DE JOGADORES DE FUTEBOL.

São Carlos - SP
2018

HENRIQUE F. LOPES

APRENDIZADO DE MÁQUINA APLICADO A PREVISÃO DE DESEMPENHO DE JOGADORES DE FUTEBOL.

Trabalho apresentado à Escola de Engenharia de São Carlos da Universidade de São Paulo para obtenção do Título de Engenheiro Eletricista com ênfase em Eletrônica.

Orientador:

Prof. Dr. Evandro L. L. Rodrigues

São Carlos - SP
2018

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da EESC/USP com os dados inseridos pelo(a) autor(a).

FF 471 Aa

Figueira Lopes, Henrique
Aprendizado de Máquina aplicado a previsão de desempenho de jogadores de futebol. / Henrique Figueira Lopes; orientador Evandro Luis Linhares Rodrigues. São Carlos, 2018.

Monografia (Graduação em Engenharia Elétrica com ênfase em Eletrônica) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2018.

1. Aprendizado de Máquina. 2. Inteligencia Artificial. 3. Cartola. 4. Futebol. 5. Modelagem. I. Título.

Eduardo Graziosi Silva - CRB - 8/8907

FOLHA DE APROVAÇÃO

Nome: Henrique Figueira Lopes

Título: "Aprendizado de máquina aplicado a previsão de desempenho de jogadores de futebol"

Trabalho de Conclusão de Curso defendido e aprovado em 25 / 06 / 19 ,

com NOTA 9,5 (NOVE, CINCO.), **pela Comissão Julgadora:**

Prof. Associado Evandro Luis Linhari Rodrigues - Orientador - SEL/EESC/USP

Prof. Associado Dennis Brandão - SEL/EESC/USP

Prof. Dr. Gustavo Enrique de Almeida Prado Alves Batista - SCC/ICMC/USP

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Rogério Andrade Flauzino

Dedicatória

Dedico esse trabalho a todos os meus familiares e a minha namorada pelo apoio e incentivo durante a realização deste projeto.

AGRADECIMENTOS

Agradeço aos meus familiares e a minha namorada pelo apoio e aos colegas que estiveram envolvidos de forma direta e indireta com a realização deste trabalho.

RESUMO

Este trabalho buscou aplicar modelos de aprendizado de máquina para a predição do desempenho de jogadores de futebol no Campeonato Brasileiro, tendo como base os dados do CartolaFC. O trabalho compreendeu todas as etapas entre extração de dados, criação de *features*, treinamento do modelo e avaliação dos resultados. Objetivou-se analisar a viabilidade de um modelo preditivo aplicado a um problema com tanta incerteza e subjetividade como o futebol. Além disso, foram treinandos modelos utilizando diferentes algoritmos, mais especificamente, os algoritmos de Regressão Linear, K-Vizinhos mais Próximos e *Gradient Boosting* foram utilizados e seu desempenho foi comparado tendo como base o conhecimento prévio sobre as vantagens e desvantagens de cada abordagem.

Palavras-Chave – Aprendizado de Máquina, Inteligência Artificial, Futebol, CartolaFC, Esporte, Modelagem.

ABSTRACT

This work aimed on applying machine learning to predict the performance of football players on the Brazilian Football League, having as data the information from CartolaFC, a famous fantasy game. The work streches from data extraction and feature creation to training machine learning models and comparing their performance. The main goal was to analyse the viability of predictive models applied to such an unpredictable and uncertain problem as football players performance. Also, three different algorithms were used, namely, Linear Regression, K-Nearest Neighbours and Gradient Boosting were used and their performance was compared using knowledge from previous works and the literature about the advantages and downsides of each approach.

Keywords – *Machine Learning, Artificial Intelligence, Soccer, Football, Sports, Modelling.*

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

"A person who never made a mistake never tried anything new."

-- Albert Einstein

PARTE I

INTRODUÇÃO

1 INTRODUÇÃO

A utilização de Inteligência Artificial (AI) e Aprendizado de Máquina (ML) têm se mostrado fundamental para diversas empresas e organizações como forma de analisar, interpretar e classificar dados. Algumas das grandes empresas de tecnologia do mundo como *Amazon, Google* [1] e *Facebook* investem bilhões de dólares nessas tecnologias e seus principais líderes já declaram acreditar que esses algoritmos irão revolucionar a tecnologia em todo o mundo. A aplicação de ML já é realidade em muitas áreas como reconhecimento de imagens [2], *chatbots* automatizados, e muitos outros.

No trabalho em questão, busca-se estudar a viabilidade de aplicação destes algoritmos na área esportiva que até o momento foi pouco explorada por pesquisadores na área de aprendizado de máquina.

1.1 Motivação

A primeira motivação do trabalho surge da observação do sucesso das técnicas de Aprendizado de Máquina para predição de resultados e para classificação em categorias, nas mais diversas áreas como é o caso para a Bolsa de Valores [3] e para recomendação de produtos em sites de compras online.

A segunda grande motivação refere-se ao conhecimento de que a área esportiva, curiosamente, não dispõe de muitas pesquisas relacionadas à predição de desempenho de jogadores, embora existam muitos dados disponíveis para que esses algoritmos sejam utilizados.

Então, como última grande motivação tem-se o *fantasy game*, CartolaFC, um jogo baseado no Campeonato Brasileiro de Futebol, no qual os participantes devem selecionar um time de onze jogadores e cuja participação no jogo real é utilizada para calcular a pontuação de cada participante na rodada. O Cartola, possui uma grande base de dados dos jogadores do Campeonato Brasileiro e o objetivo é tentar prever quais jogadores se

destacaram em cada rodada. A partir disso, observou-se uma aplicação interessante para um algoritmo preditivo na área esportiva.

1.2 Objetivo(s)

O trabalho tem como objetivo aplicar e avaliar o desempenho de diversos algoritmos de Aprendizado de Máquinas para predição do desempenho de jogadores de futebol no Campeonato Brasileiro.

1.3 Justificativas/Relevância

Acredita-se que o trabalho justifica-se como forma de demonstrar como a Inteligência Artificial pode ser aplicada para prever e analisar o desempenho de jogadores de futebol. Além disso, acredita-se que o trabalho ganha relevância científica ao apresentar uma extensa comparação entre diversos métodos de aprendizado de máquina comparando a desempenho destes quando aplicados ao problema em questão.

1.4 Organização do Trabalho

Este trabalho está dividido em 5 capítulos, incluindo esta Introdução, dispostos conforme a descrição que segue: Capítulo 2 - Embasamento Teórico: Descreve a base teórica necessária para construção e compreensão do trabalho realizado. Nessa seção serão também descritos os estudos mais relevantes, e que serviram como embasamento teórico para o projeto em questão.

Capítulo 3 - Materiais e Métodos: Discorre sobre os materiais e métodos utilizados para o projeto. Serão apresentadas as principais ferramentas, linguagens e serviços utilizados na implementação dos sistemas, bem como os métodos e etapas necessárias para alcançar o objetivo inicialmente estipulado. Finalmente, serão definidas as métricas para avaliação de resultados que serão utilizadas nas seções seguintes para mensurar o desempenho dos modelos implementados.

Capítulo 4 - Resultados: Apresenta os resultados obtidos após a execução dos códigos implementados, além de gráficos de desempenho e cálculo do desempenho dos modelos baseado nas métricas definidas na seção 3 - Materiais e Métodos.

Capítulo 5 - Discussão: Apresenta uma análise e interpretação detalhada dos resulta-

dos obtidos e do por que destes resultados, faz-se uma cuidadosa comparação dos modelos implementados, minuciosa avaliação para assegurar que os valores obtidos sejam condizentes com o esperado tendo em vista a teoria descrita na seção Embasamento Teórico.

Capítulo 6 - Conclusão: Descreve os objetivos alcançados, o conhecimento adquirido e o sucesso ou limitação da metodologia para gerar um modelo preditivo do desempenho esportivo de jogadores. Observa-se o impacto gerado pelo trabalho, o legado eventualmente possa ser deixado, finalmente, apontam-se para os problemas e dificuldades encontrados durante o seu desenvolvimento e os próximos passos no sentido de dar continuidade ao trabalho.

2 EMBASAMENTO TEÓRICO OU FUNDAMENTAÇÃO TEÓRICA

Nessa seção serão abordados os conhecimentos teóricos necessários para a compreensão do trabalho, e então serão analisadas pesquisas na área que tiverem influência na implementação desse projeto.

2.1 Base Teórica

2.1.1 Medidas Estatísticas

Aplicações de *machine learning* necessitam de grande quantidade de dados, por isso o conhecimento de métodos estatísticos é fundamental para compreender, e transformar os dados da maneira adequada.

2.1.1.1 Desvio Padrão

Desvio padrão é uma medida de dispersão em relação a média de um conjunto amostral. É muito utilizado para compreender a distribuição de uma variável [4].

$$\Theta = \sqrt[2]{\frac{1}{N} * \sum (x_i - \mu)^2}$$

2.1.1.2 Correlação

Correlação é um método estatístico para medir a relação, seja ela causal ou não, entre duas variáveis [4]. Essa medida é dada pela fórmula:

$$p_x, y = E[(X - \mu_x)(Y - \mu_y)]/\Theta_x * \Theta_y$$

Em que Θ é o desvio padrão e μ é o valor esperado da variável.

2.1.1.3 Métricas e Índices

Para compreender a desempenho de um modelo preditivo é fundamental comparar o resultado dado pelo modelo com o valor real esperado, por esta razão, diversas métricas estatísticas são utilizadas para medir a desempenho de modelos. Na tabela abaixo são apresentados alguns desses índices e como eles são calculados.

Tabela 1: Métricas de Desempenho.

Nome da Métrica	**Equação**
Erro médio absoluto	$\frac{\sum \lvert y_i - x_i \rvert}{n}$
Erro médio quadrado	$\frac{\sum (y_i - x_i)^2}{n}$
Erro mediano absoluto	$median(\lvert y_i - x_i \rvert)$

Fonte: Próprio Autor.

Existe ainda outra métrica muito utilizada para analisar a desempenho de um modelo que é a métrica R-quadrado. Esta é uma medida estatística que mede quão próxima uma predição está dos dados reais, e é dada por:

$$R_{quadrado} = \frac{VarianciaExplicada}{VarianciaTotal}$$

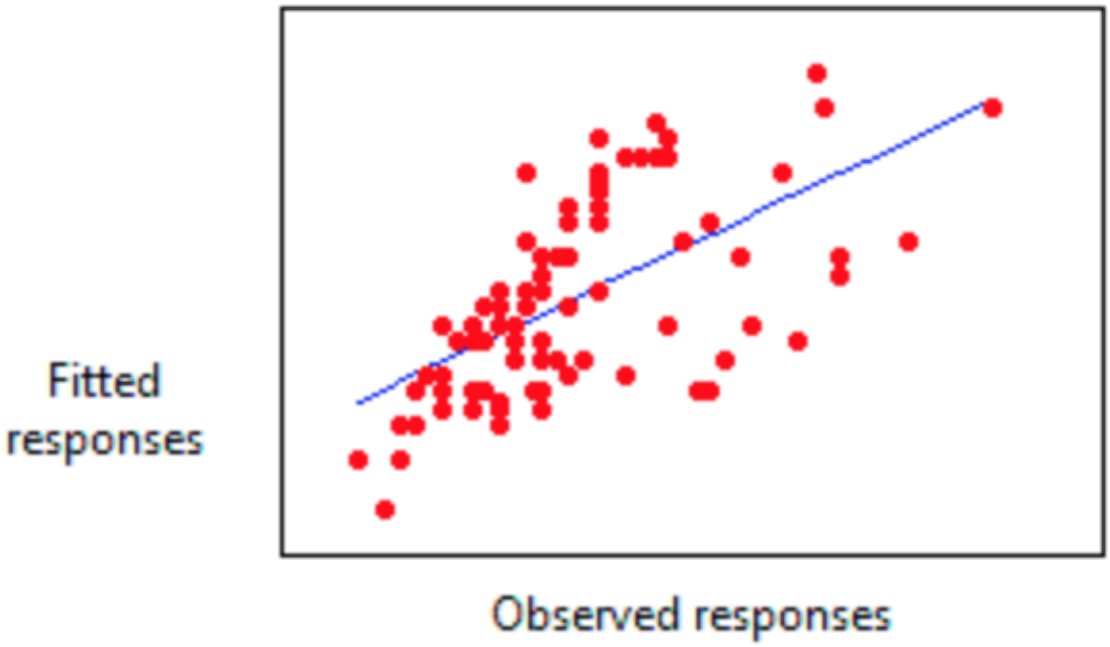
, em que variância explicada é a variância das predições do modelo e variância total é a variância real dos dados [5], como mostrado na Figura 1, na qual pode-se observar que a reta traçada não captura toda a informação dos dados observados.

2.1.2 SQL

SQL ou *Structured Query Language*, é a linguagem de consulta de bancos de dados mais utilizada em todo o mundo. Isso se deve a grande versatilidade, eficiência e simplicidade da linguagem.

A linguagem SQL é a base da grande maioria das linguagens de análise de dados em diversas linguagens de programação. Neste projeto, utilizou-se SQL para criação, junção e filtragem de dados.

Figura 1: Exemplo de diferença entre predição e dados



Fonte: Disponível em: http://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit. Acesso em 14 Nov 2017. [5]

2.1.3 Aprendizado de Máquina Supervisionado

Aprendizado de Máquina, do inglês *Machine Learning*, compreende o campo das Ciências da Computação que estuda o desenvolvimento de algoritmos preditivos baseados na análise de dados. Como descrito em [6], em um cenário típico, tem-se um resultado esperado, que deseja-se prever utilizando um conjunto de características, e utilizam-se esses dados para construção de um modelo que aprende no conjunto de treino e é capaz de prever o resultado de dados futuros, não observados

Em geral, essa decisão é usada para classificação de elementos em categorias, por exemplo, classificar fotos de animais entre diferentes espécies ou para decisões contínuas como, por exemplo, previsão do valor de ações no mercado financeiro [3].

Problemas como estes, em que existe um conjunto de treinamentos para o qual o objetivo de predição é conhecido, são chamados de Aprendizado de Máquina Supervisionado. Como definido em [6], Aprendizado de Máquina Supervisionado ocorre quando utiliza-se um conjunto de dados medidos *inputs*, com o intuito de prever os valores de saída, dado que existe uma relação entre esses dados medidos e a saída que deseja-se prever.

2.1.4 Regressão Linear

Regressão Linear é um dos métodos estatísticos mais utilizados nos últimos 30 anos e continua sendo uma das mais importantes ferramentas de modelagem [6]. A ideia do método é de dado um conjunto de entrada

$X_t = (X_1, X_2, ..., X_p)$ busca-se modelar a saída Y como $Y = A + \sum X_i * C_i$ para i de 1 a p e então para cada dado observacional, busca-se minimizar o erro entre Y_{prev} e Y_{real} por meio do ajuste dos parâmetros A e C_i [6]. E essa minimização é feita utilizando os dados de entrada e saída previamente obtidos.

2.1.5 K-Vizinhos Mais Próximos

O método dos K-Vizinhos Mais Próximos também é um método bastante antigo e intuitivo de Aprendizado de Máquina. Nesse método são utilizadas as observações dos dados de treino para achar os K vizinhos mais próximos ao dado que deseja-se prever dada uma métrica de proximidade.

Em geral, essa métrica é a distância euclidiana entre os dados os dados de entrada, ou seja, a distancia entre o elemento N_1 e N_2 com dados de entrada $(X_{11}, X_{12}, ..., X_{1n})$ e $(X_{21}, X_{22}, ..., X_{2n})$ é dado pela fórmula

$$D_{12} = \frac{1}{N} \sum (X_{1i} - X_{2i})^2 \tag{2.1}$$

.

E então, como explicado em [6], o valor previsto para uma amostra será dado por:

$$Y = \frac{1}{k} \sum (y_i) \tag{2.2}$$

Em que K são os vizinhos mais próximos, ordenados pela métrica D explicada anteriormente.

Como pode-se observar o método de K-Vizinhos Mais Próximos não depende de uma etapa de treinamento, como a Regressão Linear, isto é, não existe um ajuste dos parâmetros para minimização de erros [7]. Essa é a principal diferença entre o método o K-Vizinhos Mais Próximos e outros métodos tradicionais de aprendizado supervisionado.

2.1.6 Gradiente Boost

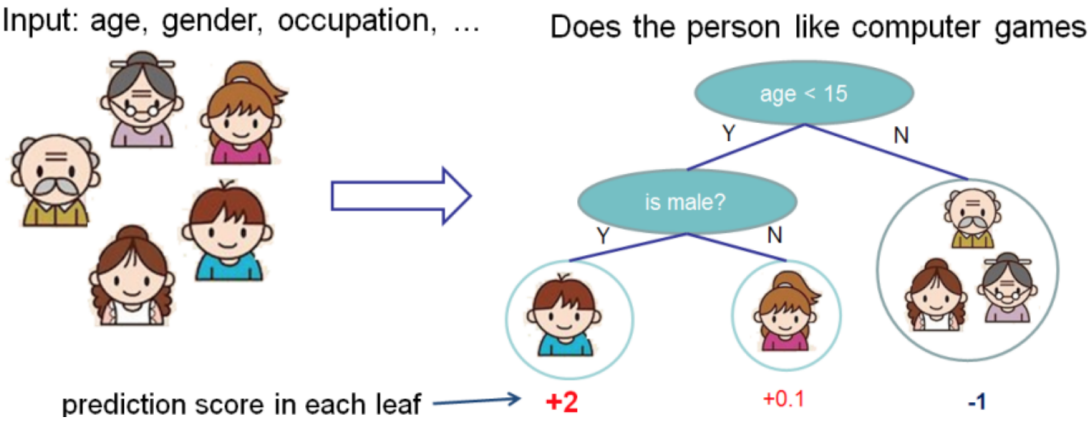
Boosting é um das ideias mais poderosas na área de Aprendizado de Máquina introduzido nos últimos vinte anos [6]. A ideia de *boosting* é combinar outros classificadores chamados de "classificadores fracos"para gerar um "comitê de classificadores". Com isso, é possível criar diversos classificadores que são qualificados para prever parte do conjunto de dados, e combinar esses classificadores para obtenção de um modelo mais robusto.

Esses classificadores podem ser qualquer modelo como regressão linear, ou k-vizinhos mais próximos mencionados anteriormente, no entanto, consagraram-se os modelos de *boosting* que utilizam árvores de decisão como classificadores fracos.

Gradient Boosting é justamente um algoritmo de *boosting* que utiliza árvores de decisão como classificadores fracos.

A Figura 2 abaixo apresenta um exemplo simples de árvore de decisão, em que pode-se observar de forma simplificada, o funcionamento deste tipo de modelagem:

Figura 2: Exemplo de árvore de decisão.



Fonte: Disponível em: ⟨https://xgboost.readthedocs.io/en/latest/model.html⟩ Acesso em 05 Fev 2018. [8]

Portanto, a partir da figura, observa-se como cada nó da árvore irá dividir o espaço N dimensional de características em 2 espaços complementares, e o erro dado por esse corte será calculado pelo número de amostras que foram classificadas de forma errada pelo corte realizado [8].

Como o *boosting* realiza a combinação de diversas árvores, serão treinados múltiplos modelos, sendo que cada uma deles irá utilizar apenas um conjunto limitado de características de forma que cada árvore aprenda a separar o espaço baseado em diversos fatores distintos. Então, o resultado do modelo será a média do valor de decisão de cada árvore

utilizada.

Esse método tem apresentado resultados bastante positivos em diversas áreas de aplicação e hoje é um dos algoritmos mais utilizados para criação de modelos preditivos.

2.2 Pesquisas Relacionadas

Em [9] foram utilizados três variações de Redes Neurais para tentar prever quais seriam os jogadores convocados para o Jogo da Estrelas da NBA. Este trabalho utilizou a seguinte metodologia. Primeiramente realizou-se a construção de um conjunto de dados a partir de dados obtidos na internet, a partir do qual foram feitas as escolhas das características a serem utilizadas variáveis do modelo, então realizou-se a avaliação de como esta escolha de características influencia o resultado preditivo dos três modelos treinados. Os algoritmos utilizados foram *Feed Forward Neural Networks'*, *Radial Basis Function Networks* e *AdaBoost*.

O conjunto de dados utilizados é mostrado na Figura 3, em que observa-se a grande quantidade de informações que são registradas para cada jogador na NBA, algo que não é tão usual no futebol.

Figura 3: Features utilizadas para previsão de desempenho na NBA.

Season	Age	Tm	Lg	Pos	G	GS	MP	FG	FGA	FG%	3P	3PA	3P%	2P	2PA	2P%	eFG%	FT	FTA	FT%
2008-09	20	LAC	NBA	C	53	13	14.5	1.8	2.8	.633	0.0	0.0		1.8	2.8	.633	.633	0.8	2.1	.385
2009-10	21	LAC	NBA	C	70	12	16.2	2.1	3.4	.605	0.0	0.0	.000	2.1	3.4	.610	.605	0.7	1.8	.375
2010-11	22	LAC	NBA	C	80	66	25.6	2.9	4.3	.686	0.0	0.0	.000	2.9	4.3	.688	.686	1.2	2.7	.452
2011-12	23	LAC	NBA	C	**66**	66	27.2	3.1	4.9	.632	0.0	0.0	.000	3.1	4.9	.634	.632	1.1	2.1	.525
2012-13	24	LAC	NBA	C	**82**	82	24.5	3.8	6.0	**.643**	0.0	0.0		3.8	6.0	**.643**	**.643**	1.2	3.0	.386
2013-14	25	LAC	NBA	C	82	82	35.0	4.2	6.3	**.676**	0.0	0.0		4.2	6.3	**.676**	**.676**	2.0	4.6	.428
2014-15	26	LAC	NBA	C	82	82	34.4	4.6	6.5	**.710**	0.0	0.0	.250	4.6	6.5	**.713**	**.711**	2.3	5.7	.397
2015-16	27	LAC	NBA	C	77	77	33.7	4.6	6.6	**.703**	0.0	0.0	.000	4.6	6.6	**.704**	**.703**	3.5	8.0	.430
2016-17 ★	28	LAC	NBA	C	81	81	31.7	5.1	7.1	**.714**	0.0	0.0	.000	5.1	7.1	**.717**	**.714**	2.5	5.2	.482
2017-18	29	LAC	NBA	C	77	77	31.5	4.8	7.5	.645	0.0	0.0		4.8	7.5	.645	.645	2.4	4.1	.580
Career			**NBA**		**750**	**638**	**28.1**	**3.8**	**5.7**	**.673**	**0.0**	**0.0**	**.091**	**3.8**	**5.7**	**.674**	**.673**	**1.8**	**4.1**	**.446**

Fonte: Disponível em: ⟨https://www.basketballreference.com/players/h/hasleud01.html⟩ Acesso em 03 Agosto 2017.

Com a seleção de dados adequada e utilizado o método *AdaBoost* o modelo conseguiu uma taxa acerto na ordem de 90%.

Em [10] utilizaram-se seis algoritmos de *machine learning* para tentar prever o resultado de diversas partidas de futebol da *Champions League*. Neste caso, os algoritmos de aprendizado utilizados foram:

- *Naive Bayes*
- *Bayeasin Network*
- *LogitBoost*
- *K-Neareste Neighboors*
- *Random Forest*
- *Neural Networks*

Quanto as características, foram selecionadas:

- Fase do time
- Histórico do confronto
- Colocação no torneio
- Número de jogadores machucados
- Saldo de Gols
- Avaliação subjetiva da qualidade do time feita por especialistas

O aspecto mais relevante dessa pesquisa foi procurar entender como uma medida subjetiva de qualidade do time poderia afetar o resultado do modelo. O melhor desempenho do modelo foi de 68% de acurácia.

Em [11] foram utilizados diversos algoritmos genéticos para predição do resultado de partidas de futebol da Liga Inglesa. O resultado desses algoritmos foi utilizado em conjunto como o objetivo de obter um melhor resultado global. Com esta finalidade, a junção dos diversos algoritmos foi feita de aplicando diferentes processos, como votação, média, ou análise estatística procurando identificar qual método apresentaria o melhor resultado.

Comparando-se os resultados obtidos com a aplicação de algoritmos genéticos aos resultados obtidos utilizando Redes Neurais e ambos alcançaram um índice de acerto próximo a 70%.

Em [12], Kou-Yuan Huang utilizou uma Rede Neural para tentar prever os resultados da Copa do Mundo de 2006. Nessa pesquisa, foi feito um estudo de como a variação de parâmetros do algoritmo poderia afetar o resultado do modelo. Essa variação foi obtida ao utilizar as características originais e combina-las segundo conhecimento prévio da relação entre elas. O resultado final obtido foi de 73,6% de acurácia.

Finalmente, em [13], Henrique Gomide realizou análise exploratória dos dados do jogo Cartola, jogo online de simulação baseado no campeonato Brasileiro de Futebol, para tentar prever o desempenho dos jogadores e o placar dos jogos, de modo a melhorar o índice de acerto nas escolhas no jogo. Como *features* foram utilizados os *scouts* básicos do jogo cartola, e o modelo construído utilizou redes neurais.

3 MATERIAIS E MÉTODOS

Nesta seção serão apresentadas as ferramentas utilizadas no trabalho, bem como a sequência de etapas seguida durante a execução deste projeto para alcançar os objetivos almejados.

3.1 Materiais

3.1.1 Python

A linguagem de programação Python é uma linguagem muito versátil e de código aberto, isto é, permite licenciamento livre para qualquer produto feito com essa linguagem. Estes aspectos contribuem para que esta seja uma das linguagens mais utilizadas tanto para pesquisa quanto nas grandes empresas.

Pela sua vasta utilização, é muito fácil encontrar fóruns de ajuda e bibliotecas bastante completas e atualizadas, facilitando a codificação das funcionalidades desejadas.

3.1.2 PostGreSQL

PostGreSQL é um dos mais robustos sistemas de organização de bancos de dados baseados na linguagem SQL. A escolha da linguagem SQL se deu por ser uma linguagem bastante utilizada e bastante intuitiva. Já a escolha do *PostGres* se deu pela flexibilidade, e pela possibilidade de inserção de vetores e dados no formato JSON, que serão utilizados no desenvolvimento do nosso trabalho.

3.1.3 Amazon Web Services (AWS)

O AWS é um serviço de computação em nuvem com diversas funcionalidades como bancos de dados, computação distribuída, armazenamento de sites e monitoramento de

máquinas. No caso particular deste trabalho, foram utilizados as tecnologias de armazenamento de bancos de dados (RDS) e o serviço de computação em nuvem (ECS).

O sistema RDS permite não só que o banco de dados seja acessado de qualquer computador, mas também oferece serviço de backup automático, monitoramento e alta disponibilidade dos dados, sendo bastante vantajoso para armazenamento dos dados utilizados.

O serviço de computação em nuvem permite que a análise de dados e o treinamento de modelos seja feito em nuvem, tornando fácil a ampliação da capacidade computacional disponível, bem como garantindo que o modelo possa ser treinado independentemente do computador utilizado para desenvolvimento do código.

3.1.4 SKLearn (SK), TensorFlow (TF), XGBoost (XGB), Pandas

As bibliotecas SK, TF e XGB são bibliotecas de Python que encapsulam algoritmos e funções de *machine learning*.

A biblioteca SK possui funções de linearização, normalização, redução de dimensões e outras.

Por sua vez, a biblioteca TF encapsula diversas funções necessárias para construção de redes neurais. Nessa biblioteca é possível definir facilmente diversos parâmetros, como, número de camadas, nós em cada camada, função de ativação de cada camada, entre muitos outros. Além disso, ainda é possível utilizar esses objetos como base para treinamento do modelo.

De forma semelhante, a biblioteca XGB encapsula as funções necessárias para execução do algoritmo *Extreme Gradient Boost*, que também foi utilizado no projeto.

Finalmente, a biblioteca de Python Pandas, que também é uma biblioteca que encapsula diversas funções necessárias para análise e transformação de dados, e que é intensamente utilizada em problemas complexos de análise de dados em Python.

3.2 Métodos

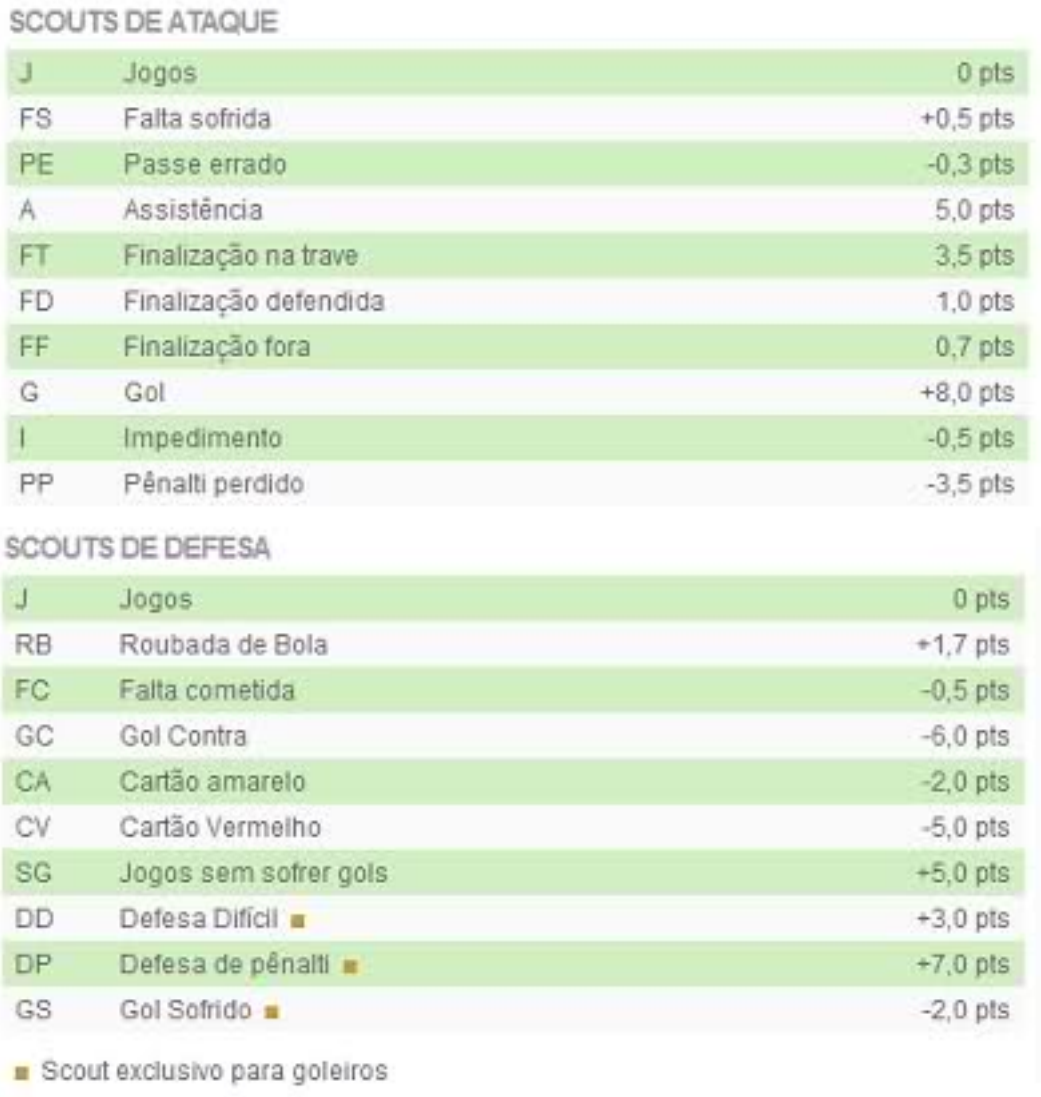
O desenvolvimento do projeto foi trabalho em quatro partes, sendo elas: Construção e Organização do Banco de Dados, Desenvolvimento de características e transformação dos dados, Treinamento e Refinamento dos modelos e, Geração e Comparação dos Resultados.

3.2.1 Construção e Organização dos dados

Como descrito na seção introdutória deste trabalho, a base de dados e sistema de predição será construído sobre a base de dados do jogo Cartola FC.

Os principais dados presentes no jogo são referentes a desempenho de cada jogador em algumas categorias definidas pelo jogo. As categorias avaliadas e seu respectivo valor no jogo são mostrados em seguida, na Figura 4, esse conjunto de categorias é conhecido como *Scouts*:

Figura 4: Categorias Avaliadas e Pontuação.



SCOUTS DE ATAQUE

J	Jogos	0 pts
FS	Falta sofrida	+0,5 pts
PE	Passe errado	-0,3 pts
A	Assistência	5,0 pts
FT	Finalização na trave	3,5 pts
FD	Finalização defendida	1,0 pts
FF	Finalização fora	0,7 pts
G	Gol	+8,0 pts
I	Impedimento	-0,5 pts
PP	Pênalti perdido	-3,5 pts

SCOUTS DE DEFESA

J	Jogos	0 pts
RB	Roubada de Bola	+1,7 pts
FC	Falta cometida	-0,5 pts
GC	Gol Contra	-6,0 pts
CA	Cartão amarelo	-2,0 pts
CV	Cartão Vermelho	-5,0 pts
SG	Jogos sem sofrer gols	+5,0 pts
DD	Defesa Difícil ■	+3,0 pts
DP	Defesa de pênalti ■	+7,0 pts
GS	Gol Sofrido ■	-2,0 pts

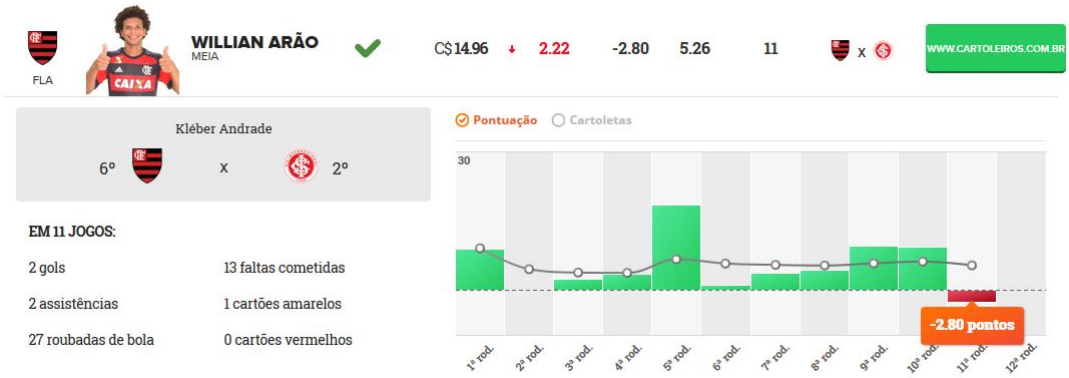
■ Scout exclusivo para goleiros

Fonte: Disponível em: ⟨https://cartolafc.globo.com/time⟩ Acesso em 20 Out 2017.

Dessa maneira, a cada partida disputada no campeonato o jogador terá seu desempenho avaliado e a pontuação definida. Como mostrado na Figura 5, na qual pode-se observar a variação de pontos do jogador nas diferentes rodadas, a variação de preço do

jogador dado seu desempenho, entre outras características.

Figura 5: *Scout* Exemplo de um Jogador.



Fonte: Disponível em: ⟨https://cartolafc.globo.com/time⟩ Acesso em 20 Out 2017.

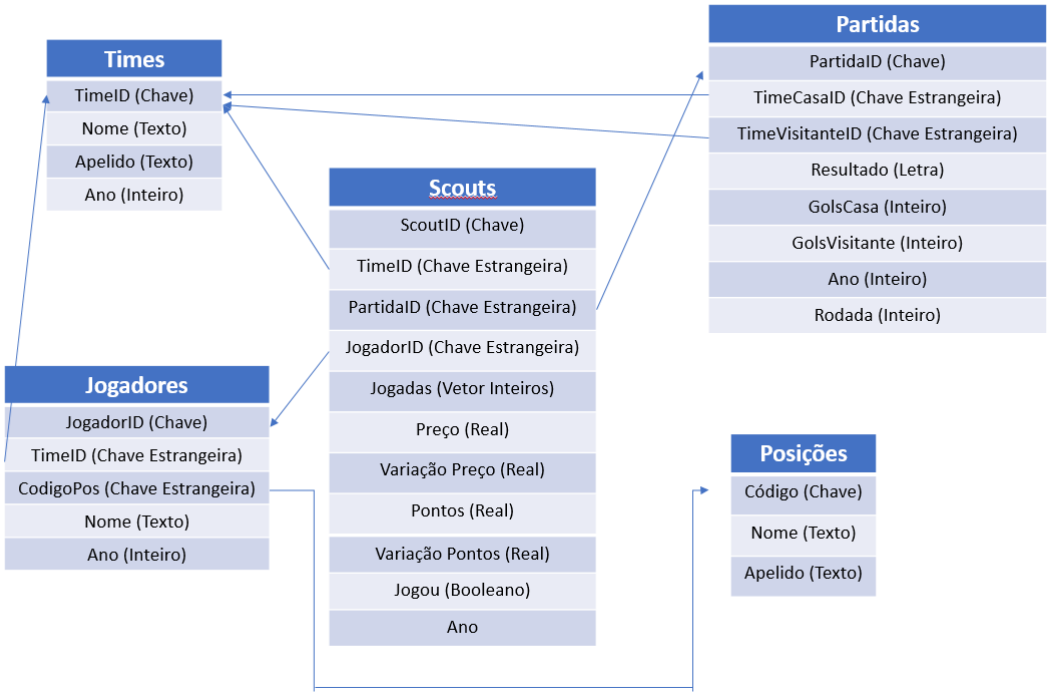
Além desses dados, é possível obter dados como a pontuação dos times no campeonato, os próximos confrontos de cada time, os resultados dos últimos jogos de cada time. Todos esses dados serão utilizados pelo nosso modelo por isso é essencial que eles estejam presentes no banco de dados.

Analisando os dados existentes, foi possível definir a estrutura de dados das tabelas presentes no banco de dados. No diagrama da Figura 6 podemos ver os dados das tabelas e como estes se relacionam, a partir da imagem é possível observar que para cada *Scout* seria possível obter informações sobre o seu time, o time adversário, a sua posição, dados que podem ser bastante uteis como entradas do modelo.

Para obtenção desses dados, o jogo oferece um serviço baseado no sistema de API's com os seguintes *endpoints*:

- ⟨https://api.cartolafc.globo.com/mercado/status⟩ - Status do mercado
- ⟨https://api.cartolafc.globo.com/mercado/destaques⟩ - Lista dos jogadores mais escalados
- ⟨https://api.cartolafc.globo.com/rodadas⟩ - Lista das rodadas do campeonato
- ⟨https://api.cartolafc.globo.com/partidas⟩ - Próximas partidas do campeonato
- ⟨https://api.cartolafc.globo.com/clubes⟩ - Lista de clubes
- ⟨https://api.cartolafc.globo.com/atletas/mercado⟩ - Lista de todos os jogadores
- ⟨https://api.cartolafc.globo.com/atletas/pontuados⟩ - Pontuação da rodada em andamento

Figura 6: Diagrama de Dados



Fonte: Próprio Autor.

- ⟨https://api.cartolafc.globo.com/pos-rodada/destaques⟩ - Time que mais pontuou na rodada anterior
- ⟨https://api.cartolafc.globo.com/times?q=[nomedotime]⟩ - Busca geral de times
- ⟨https://api.cartolafc.globo.com/time/slug/[slugdotime]⟩ - Busca informações de um time específico
- ⟨https://api.cartolafc.globo.com/time/slug/[slugdotime]/[rodada]⟩ - Busca informações de um time específico por rodada
- ⟨https://api.cartolafc.globo.com/esquemas⟩ - Lista os esquemas táticos

Portanto, a atualização do banco de dados é realizada utilizando essas *APIs* para obter informações atualizadas de cada rodada do campeonato.

No entanto, a maioria destas informações está disponível para a rodada atual e não existe nenhum *endpoint* que retorne as informações de rodadas anteriores e anos anteriores.

Por isso, para construção do histórico dos jogadores em outros anos, obteve-se planilhas com informações completas de jogadores, times e partidas a partir do ano de 2014 que foram disponibilizadas por [13].

Com acesso a estes dados, foram desenvolvidos códigos para leitura, mapeamento e conversão destes dados para a estrutura do banco de dados previamente definida e, só então, foi possível fazer a inserção destes dados no banco.

3.2.2 Desenvolvimento de características e transformação dos dados

Uma das principais qualidades de algoritmos de *Machine Learning* é serem agnósticos ao contexto dos dados. Esses algoritmos são apenas operações matemáticas realizadas sobre números com o intuito de minimizar o erro do modelo em relação ao resultado desejado, como explicado na seção 2 - Embasamento Teórico.

Por isso, é necessário que os dados que serão utilizados pelo modelo sejam convertidos em valores numéricos que de alguma forma descrevam aquela grandeza.

Para o caso em questão, essa conversão é feita por meio de *queries* SQL que não só conseguem extrair informações do banco de dados, mas são capazes de transformar essa informação em um dado numérico. A seguir está um exemplo de uma query capaz de extrair a 'fase' do time nos últimos cinco jogos, isto é, quantos jogos esse time ganhou, empatou e perdeu.

Observa-se então que este valor já está traduzido em um número seguindo a fórmula:

$Fase = 3 * V + 1 * E + 0 * D$, em que V, E e D representam as vitórias, empates e derrotas do time nos jogos recentes.

Na tabela abaixo, são apresentados alguns exemplos de informações utilizadas na modelagem do sistema preditivo.

Tabela 2: Exemplo de Features Utilizadas para Modelagem.

Nome da Feature	**Categoria**
Média de Pontos do Time	Contínua
Média de Pontos do Oponente	Contínua
Média de Pontos do Jogador	Contínua
Média de Jogadas do Jogador	Contínua
Visitante	Booleana
Média de Gols do Time	Contínua
Média de Gols Tomados pelo Time	Contínua
continua na próxima página	

Tabela 2 – continuação da pagina anterior

Nome da Feature	Categoria
Média de Gols do Oponente	Contínua
Média de Gols Tomados pelo Oponente	Contínua
Média de Gols do Time em Casa	Contínua
Média de Gols Tomados pelo Time em Casa	Contínua
Média de Gols do Oponente em Casa	Contínua
Média de Gols Tomados pelo Oponente em Casa	Contínua
Média de Gols do Time Fora	Contínua
Média de Gols Tomados pelo Time Fora	Contínua
Média de Gols do Oponente Fora	Contínua
Média de Gols Tomados pelo Oponente Fora	Contínua
Razão entre Gols Feitos pelo Time e Tomados pelo oponente	Contínua
Taxa de Participação do Jogador nos Gols do Time	Contínua
Desvio Padrão de Pontos do Jogador	Contínua
Desvio Padrão de Pontos do Time	Contínua
Posição do Jogador	Categórica
Número de Jogos sem Sofrer Gols	Discreta
Média de Faltas Sofridas	Contínua

Fonte: Próprio Autor.

Além do desafio de converter informações dos mais diversos tipos em dados numéricos, existe o desafio de saber se aquela característica passada para o modelo realmente tem influência na saída desejada. Por exemplo, um indivíduo que assiste muitos jogos sabe que em geral o seu time tem melhor desempenho quando joga em seu estádio, mas será que isso tem impacto no desempenho de cada jogador?

Para entender quais elementos são revelantes para o modelo, realizou-se uma análise extensiva dos dados em que buscou-se compreender de forma quantitativa como algumas características influenciam a desempenho de cada jogador. Nessa análise, mediu-se a correlação de cada característica com a desempenho do jogador, mas também buscou-se compreender como a combinação de algumas características influenciam a desempenho desse jogador. Os resultados dessas análises, apresentados na seção de resultados são fundamentais para criação de um modelo eficiente.

3.2.3 Treinamento e Refinamento do modelo

Após a decisão sobre quais dados serão fornecidos para o modelo, é necessário preparar esses dados para treinamento. Como a quantidade de dados é muito grande, é comum que existam inconsistências e valores faltantes nos dados, que podem prejudicar o treinamento do modelo. Por esta razão, existe uma etapa de pré-processamento, na qual é preciso decidir sobre eliminar dados inconsistentes e/ou preencher dados faltantes. Por exemplo, no estudo em questão foram eliminados do treinamento todos os dados cuja pontuação final, não correspondiam a soma das pontuações das jogadas, visto que estes dados poderiam gerar inconsistências no modelo.

Além disso, alguns algoritmos dependem de algumas condições para terem uma boa desempenho, por exemplo, algoritmos de regressão linear tem melhor desempenho se as características estiverem normalizadas. Por isso, a transformação dos dados nesse tipo de situação também é feita na etapa de pré-processamento.

Com o dataset corrigido e uniformizado é necessário definir os parâmetros do modelo. Cada algoritmo possui uma série de parâmetros a serem ajustados.

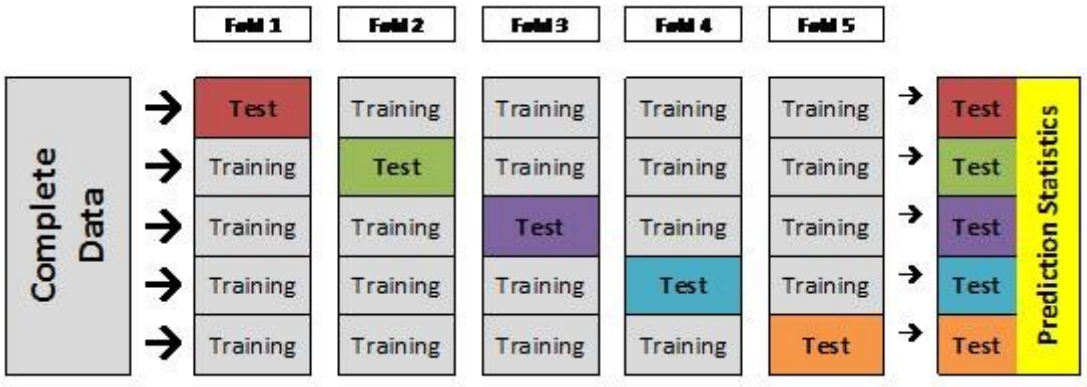
Em geral, é impossível definir de antemão qual o melhor conjunto de parâmetros para um algoritmo, já que isto depende muito do conjunto amostral, do tipo de resultado esperado, do tempo de treinamento disponível e outros elementos que não podem ser calculados antes do treinamento do modelo.

Tendo em vista esta dificuldade, a metodologia aplicada no trabalho foi a de gradiente de parâmetros. Essa metodologia, permite que sejam definidos alguns possíveis valores para cada parâmetro, dado o conhecimento do efeito daquele parâmetro no treinamento. Esses parâmetros são variados em sucessivas etapas de treinamento. Então com esses modelos calculados, obtém-se os modelos com melhor desempenho, e, consequentemente, podendo-se definir os melhores parâmetros.

Outra etapa importante da modelagem é a etapa de validação cruzada. A etapa de validação cruzada consiste em dividir as amostras em diversos subconjunto de amostras e realizar o treinamento do modelo diversas vezes, treinando o modelo em alguns elementos do subconjunto e avaliando em outro subconjunto, como mostra a Figura 7

Esse processo é de extrema importância, pois garante que a desempenho medida não seja específica de um conjunto amostral e sim consistente para diversos espaços amostrais diferentes.

Figura 7: Validação Cruzada



Fonte: Disponível em: ⟨https://www.datasciencecentral.com/profiles/blogs/how-to-train-a-final-machine-learning-model⟩ Acesso em 14 de Nov 2017. [**?**]

Para que seja possível classificar os modelos segundo seu desempenho, é necessário que se estabeleça uma série de métricas que meçam a desempenho do modelo. Embora o modelo seja treinado visando minimizar o erro médio sobre as amostras, em muitos casos, o melhor modelo não é aquele que possui menor erro médio. Neste trabalho, por exemplo, o objetivo final é minimizar o erro na predição do melhor jogador de cada posição, de forma que seja possível escalar um time forte, baseado na predição do modelo.

Embasado por esse raciocínio de maximização de resultados, foram definidos os seguintes índices de desempenho:

Tabela 3: Índices de Desempenho do Modelo

Nome do Índice	**Descrição**
Soma de Pontos Real	Soma de pontos da desempenho real dos jogadores
Soma de Pontos Previsto	Soma de pontos previsto pelo modelo para os jogadores
Desvio Padrão Real	Desvio Padrão de pontos da desempenho real dos jogadores
Desvio Padrão Previsto	Desvio Padrão de pontos previstos pelo modelo para os jogadores
Erro Médio Absoluto	Erro médio absoluto da diferença os valores previstos pelo modelo e o valor real do jogador
Erro Médio Quadrado	Erro médio quadrado da diferença entre os valores previstos pelo modelo e o valor real do jogador
Erro Mediano Absoluto	Mediana do erro amostral entre os valores previstos pelo modelo e o valor real obtido pelo jogador

continua na próxima página

Tabela 3 – continuação da pagina anterior

Nome do Índice	Descrição
Métrica R2	Medida Estatística de quanto a variância de um conjunto de amostras é explicada pelo modelo preditivo

Fonte: Próprio Autor.

Essas métricas foram utilizadas para avaliar o modelo preditivo como um todo, mas também para avaliar os agrupamentos e filtros mais específicos que melhor descrevem a desempenho real do modelo, esses agrupamentos são descritos na tabela a seguir:

Tabela 4: Agrupamentos Utilizados para avaliar a desempenho do Modelo

Nome da Agrupação	Descrição
Melhores Jogadores	Seleção dos jogadores com maior pontuação para entender como o modelo consegue prever os destaques da competição.
Melhores Jogadores Por Rodada	Seleção dos melhores jogadores por rodada para entender como o modelo performa em diferentes rodadas.
Melhores Jogadores Por Posição	Seleção de jogadores por posição para entender a qualidade do modelo para as diferentes posições dos jogadores.
Melhor Time Possível	Escolha do melhor time previsto pelo modelo para entender quantos pontos o modelo realmente faria em uma rodada.
Jogadores removendo valores atípicos	Remoção dos 10 % Jogadores com melhores e piores pontuações para entender a desempenho do modelo.

Fonte: Próprio Autor.

4 RESULTADOS

Na seção de resultados serão inicialmente apresentadas métricas sobre a execução e implementação do sistema, como tempo de cálculo de *features*, espaço de armazenamento utilizado, capacidade computacional e custos exigidos. Então serão apresentadas as análises quantitativas e qualitativas realizadas com base nos dados do Cartola. Essas análises são fundamentais para criação de *features* e compreensão do problema abordado. Finalmente, será realizada a comparação de desempenho de diferentes algoritmos de Aprendizado de Máquina tendo em vista os índices de desempenho definidos na seção de Materiais e Métodos.

4.1 Métricas de Execução

4.1.1 Armazenamento

O armazenamento de dados foi feito em um banco relacional PostGreSQL e ocupou 19.8 GB, sendo que esse espaço foi utilizado para armazenamento de 6 Posições de Jogadores, 27 Times, 1510 Partidas, 3810 Jogadores e 104680 *Scouts*. Além disso, o armazenamento das *features* calculadas para cada *scout* foi feito em arquivos no formato '.csv' que necessitaram um espaço de 1 GB.

4.1.2 Tempo de Execução

O modelo treinado utilizou 53 *Features*, sendo que cada uma delas foi calculada para 4 espaços de tempo (últimas 1, 10, 20 e 30 rodadas) totalizando 212 Features calculadas por *Scout*. Como existem 104680 *Scouts* foram calculadas aproximadamente 22,2 milhões de *Features*. O tempo médio medido para cálculo de *features* foi de 0.4 segundos, e utilizando processamento em nuvem, foi possível fazer o cálculo de 10 *features* em acesso paralelo ao banco de dados, além disso como as consultas são similares para a maioria dos *Scouts*, elas foram realizadas em blocos de 1000 *Scouts*. Considerando a sequência de operações,

o tempo necessário para o cálculo das *features* foi de aproximadamente

$$t = (22.2 * 10^6 * 2)/(1000 * 10) = 4400[s] \tag{4.1}$$

Para treinamento do modelo o tempo médio medido foi de 90s. No entanto, com o intuito de encontrar o melhor conjunto de parâmetros de treino, o algoritmo de treino foi executado diversas vezes, variando a parametrização utilizada.

A tabela a seguir mostra os parâmetros variados e os valores testados para cada parâmetro.

Tabela 5: Parâmetros XGBoost

Parâmetros	**Valores Utilizados**
Learning Rate	(0.1, 0.2, 0.3)
Gamma	(0, 0.5, 1)
Max Depth	(4, 5, 6)
SubSample	(0.7, 0.8, 0.9)
ColSample	(0.7, 0.8, 0.9)
Min Child Weight	(1, 3, 6)
Lambda	(0, 0.5, 1)
Alpha	(0, 0.5, 1)

Fonte: [8]. Acesso em 05 Fev 2018.

Então, utilizando o método de calibragem de parâmetros descrito na seção de Materiais e Métodos, obteve-se 33 combinações de parâmetros que foram utilizadas em sucessivas etapas de treinamento, o que resultou em um período de treinamento total de 29700 s = 8h:15min.

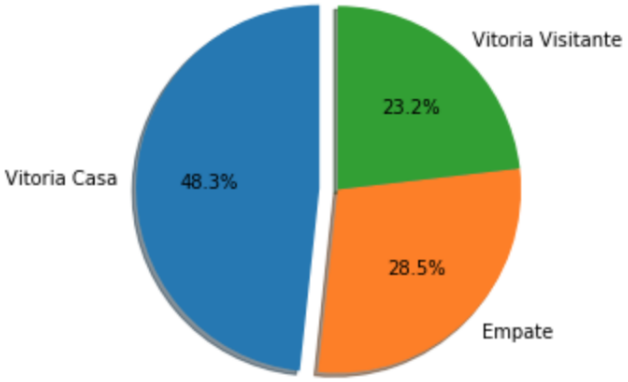
4.2 Resultados Analíticos

Nessa seção serão apresentados os resultados das análises realizadas sobre os dados armazenados, e que permitem a compreensão de aspectos e relações interessantes entre os dados.

Para a análise foram utilizados 39404 amostras, sendo que cada amostra representa um jogador, em uma rodada do campeonato.

A primeira análise realizada teve o intuito de compreender o efeito do time jogar em casa ou como visitante na desempenho dos seus atletas. Para isso, agregou-se os jogadores entre jogadores dentro de casa e fora de casa e mediu-se a porcentagem de vitórias como mostrado na Figura 8:

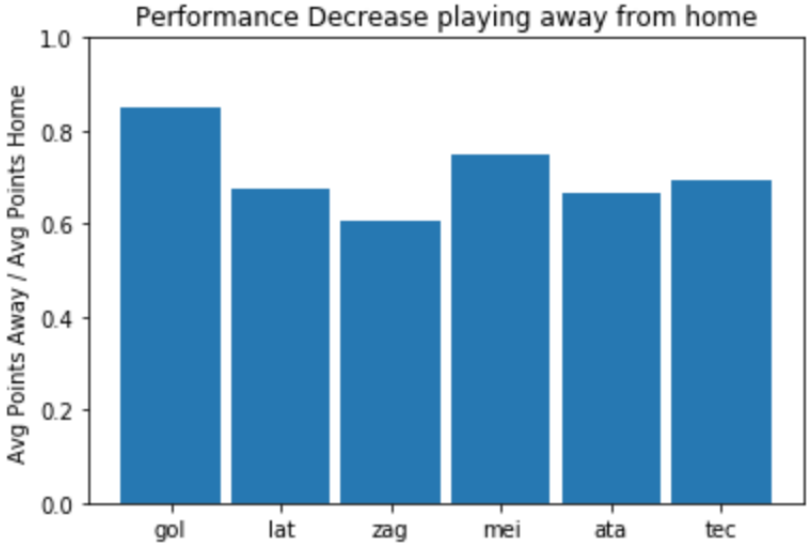
Figura 8: Aproveitamento em Casa/Fora.



Fonte: Próprio Autor.

Além disso, agrupou-se os jogador por posição para medir a influência de jogar em casa ou fora de casa para cada posição, obtendo-se o gráfico da Figura 9:

Figura 9: Desempenho por Posição em Casa/Fora



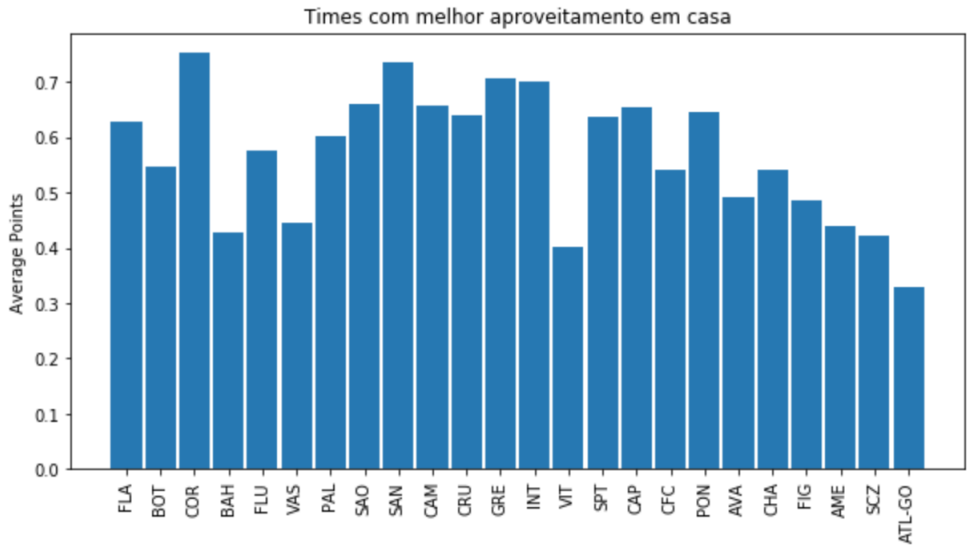
Fonte: Próprio Autor.

Em seguida, realizou-se uma análise dos melhores times e de como o desempenho do time influência a desempenho individual dos jogadores desse time.

Primeiramente obteve-se o aproveitamento de cada time do campeonato dentro e fora de casa como mostrado nos gráficos da Figura 10:

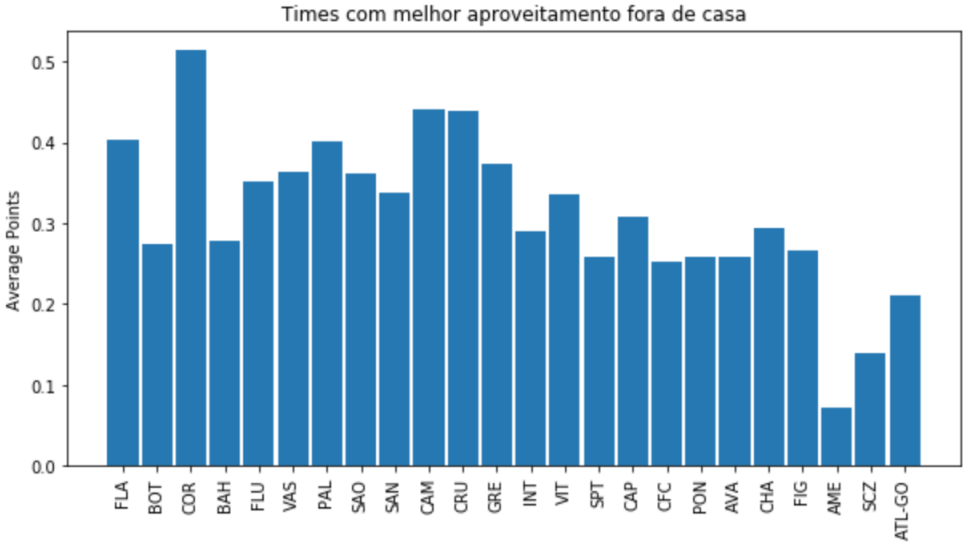
Figura 10: Desempenho

(a) Em casa



Fonte: Próprio Autor.

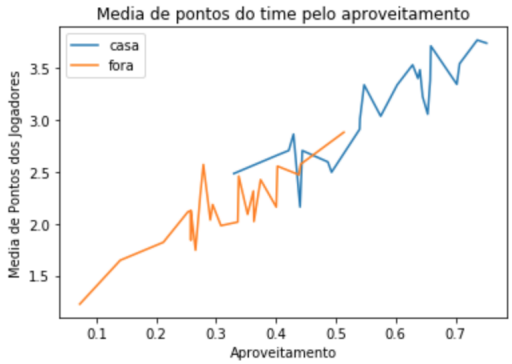
(b) Fora de casa



Fonte: Próprio Autor.

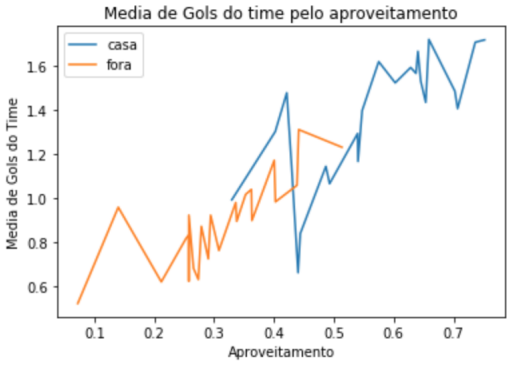
Figura 11: Gols e Pontos em função do aproveitamento

(a) Média de Pontos por aproveitamento.



Fonte: Próprio Autor.

(b) Média de Gols por aproveitamento.



Fonte: Próprio Autor.

Posteriormente, obteve-se a relação entre a média de pontos de cada jogador do time e a desempenho do time. O mesmo procedimento foi realizado para obter a relação entre o número de gols do time e seu aproveitamento. Ambos são mostrados na Figura 11

Finalmente, realizou-se uma análise separando jogadores por posição para compreender como a posição do jogador influência seu desempenho.

Então obteve-se a média de pontos e a média de custo dos jogadores para cada posição. Em seguida, foi realizado o mesmo procedimento, mas apenas para os 10% melhores de cada posição, ambos os procedimentos são mostrados na Figura 12.

Repetiu-se o procedimento, porém dividindo os atletas em jogos em que seu time venceu, empatou ou perdeu, para compreender a influência do resultado do jogo na pontuação dos jogadores, como mostrado na Figura 13.

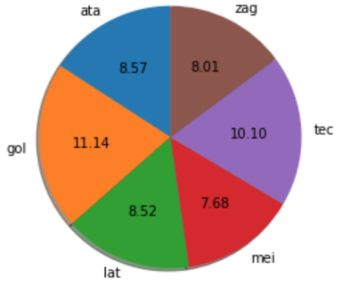
Para completar, realizou-se uma análise das jogadas que mais influenciam a pontuação de cada posição para entender quais estatísticas devem ser mais valorizadas para cada posição, como demonstram os gráficos em 14:

4.3 Resultado do treinamento

Após obtenção dos resultados das análises dos dados, realizou-se um estudo sobre os resultados dos modelos treinados. Para isso, foram utilizadas as métricas de análise de modelo definidas na seção de Materiais e Métodos aplicadas aos dados dos jogadores entre as rodadas 15 e 38 do ano de 2017. Essa escolha deu-se para evitar que o caráter temporal dos dados gerasse um resultado não representativo, e portanto utilizou-se todas

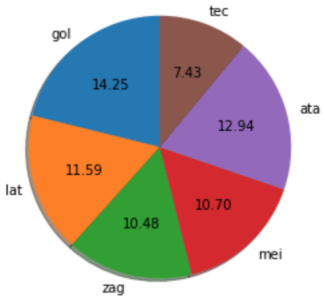
Figura 12: Pontos e Preços Por Posição

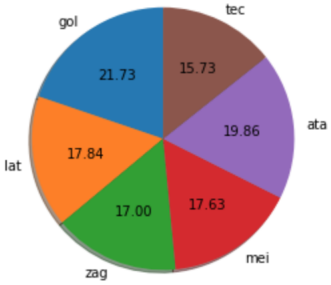
(a) Todos os jogadores



Fonte: Próprio Autor.

(b) 10% melhores jogadores

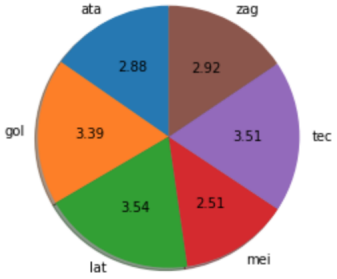




Fonte: Próprio Autor.

Figura 13: Média de Pontos por Posição

(a) Vitórias



Fonte: Próprio Autor.

(b) Empates

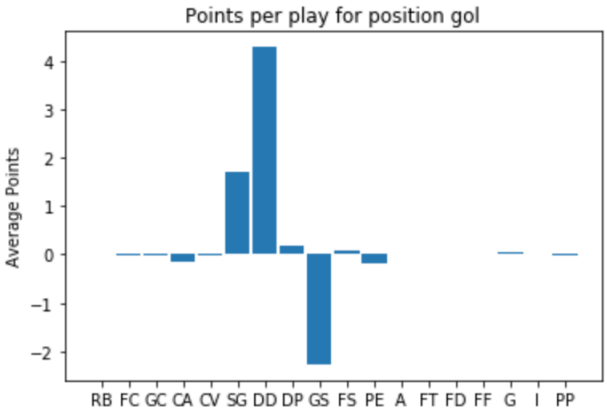
Fonte: Próprio Autor.

(c) Derrotas

Fonte: Próprio Autor.

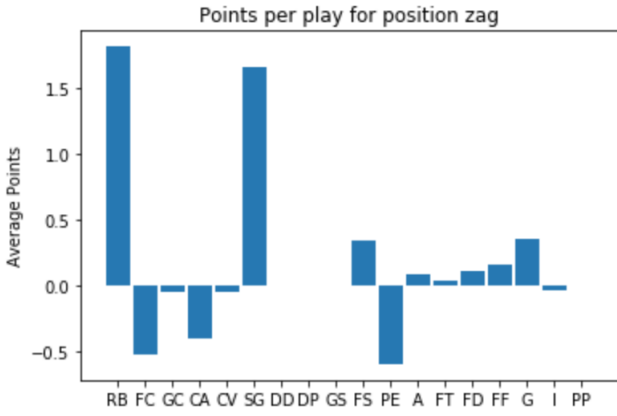
Figura 14: Distribuição de Pontos por Jogada

(a) Goleiros



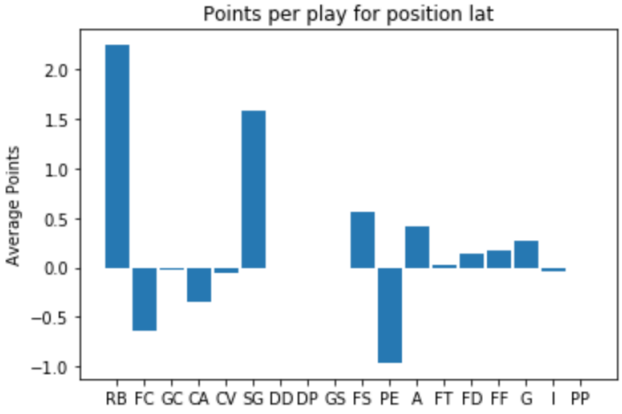
Fonte: Próprio Autor.

(b) Zagueiros



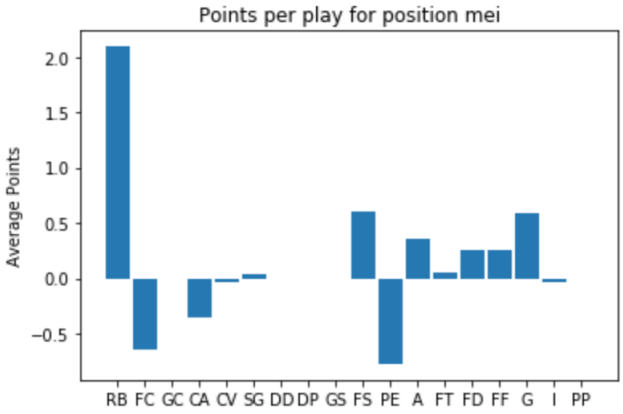
Fonte: Próprio Autor.

(c) Laterais



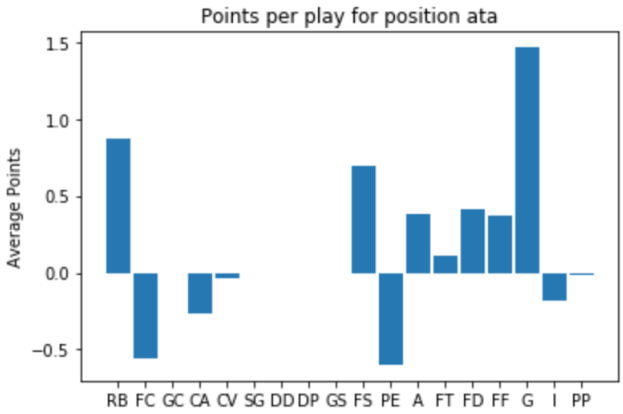
Fonte: Próprio Autor.

(d) Meias



Fonte: Próprio Autor.

(e) Atacantes



Fonte: Próprio Autor.

as rodadas antes da décima quinta para treinamento e as demais para avaliação. Assim, utilizou-se o modelo para obter um resultado de pontuação esperada para cada jogador e a partir dessa pontuação esperada obteve-se as estatísticas de desempenho de cada modelo, como é mostrado a seguir.

Primeiramente fez-se um levantamento estatístico para os jogadores presentes na base de testes e o resultado é mostrado abaixo, em que buscou-se observar como a métrica de erro absoluto e erro médio quadrado (*mean-abs-error*, *root-mean-sqrt-error*) poderiam ser comparadas entre os modelos já que a análise foi feita para os dados presentes:

Figura 15: Métricas de predição do modelo

model_name	pred_avg	pred_dev	desired_avg	desired_dev	mean_abs_error	root_mean_sqrt_error	r2_score
knn	2.552738	0.662741	3.008456	4.205884	3.198433	4.181339	0.011638
xgb	2.875175	1.149713	3.008456	4.205884	3.244436	4.228077	-0.010581
linear	2.753454	0.909043	3.008456	4.205884	3.210078	4.183414	0.010656

Fonte: Próprio Autor.

Em um segundo momento, limitou-se a análise apenas aos 100 melhores jogadores e mediu-se os resultados preditivos. Esses 100 jogadores foram ordenados tanto pela pontuação real deles quanto pela pontuação prevista para eles, e portanto obteve-se as duas tabelas abaixo, em que buscou-se observar a capacidade do modelo de selecionar os melhores jogadores, o que pode ser feito pela análise da média esperada e da média real (*pred-avg*, *desired-avg*) obtida para cada modelo.

Figura 16: Métricas de predição do modelo para os 100 melhores jogadores

model_name	pred_avg	pred_dev	desired_avg	desired_dev	mean_abs_error	root_mean_sqrt_error	r2_score
knn	2.795606	0.701793	12.982	2.690107	10.186394	10.585782	-14.484843
xgb	3.231616	1.403566	12.982	2.690107	9.750384	10.268055	-13.569252
linear	3.107444	0.999956	12.982	2.690107	9.874556	10.294106	-13.643274

Fonte: Próprio Autor.

Em seguida, foram retirados da análise os jogadores cujo desempenho estava entre os 10% piores ou melhores desempenhos e avaliou-se os demais jogadores, obtendo-se o resultado da Figura 18 em que buscou-se observar se a variância dos valores previstos se aproximava da variância real (*pred-avg*, *desired-avg*) quando os *outliers* fossem removidos.

Figura 17: Métricas de predição do modelo para as 100 previsões mais altas

model_name	pred_avg	pred_dev	desired_avg	desired_dev	mean_abs_error	root_mean_sqrt_error	r2_score
knn	3.890093	0.199547	4.648	4.851896	4.105015	4.905371	-0.022164
xgb	5.720151	1.190270	4.112	5.222304	4.633185	5.733260	-0.205255
linear	4.752859	0.518814	4.290	5.143180	4.030736	5.100733	0.016438

Fonte: Próprio Autor.

Figura 18: Métricas de predição do modelo sem jogadores destaques

model_name	pred_avg	pred_dev	desired_avg	desired_dev	mean_abs_error	root_mean_sqrt_error	r2_score
knn	2.555834	0.678103	3.06876	4.130302	3.133962	4.111173	0.009241
xgb	2.888036	1.177918	3.06876	4.130302	3.194283	4.171412	-0.020006
linear	2.769738	0.922541	3.06876	4.130302	3.150604	4.114902	0.007443

Fonte: Próprio Autor.

Finalmente, para cada rodada presente nos dados de teste foram selecionados os melhores jogadores para formação de uma escalação completa, e mediu-se o desempenho do modelo nessa situação, que corresponde a situação real caso o modelo fosse utilizado para definir a escalação para o jogo, como mostrado nas Figuras 19 e 20 e utilizou-se esses dados para estimar o desempenho real dos diferentes modelos, caso fossem utilizados no jogo, o que foi feito observando a média real (*desired-avg*) dos dados dado que foram selecionados os jogadores com melhor média prevista (*pred-avg*).

Figura 19: Métricas de predição do modelo para os melhores times

model_name	pred_avg	pred_dev	desired_avg	desired_dev	mean_abs_error	root_mean_sqrt_error	r2_score
knn	2.672245	0.418508	12.880357	1.710741	10.208112	10.414369	-92.032644
xgb	2.975223	0.615070	12.880357	1.710741	9.905134	10.106724	-85.569138
linear	2.868851	0.404830	12.880357	1.710741	10.011506	10.208875	-87.625019

Fonte: Próprio Autor.

Além das métricas quantitativas do modelo, foram traçados histogramas para compreender a distribuição da pontuação prevista de cada algoritmo em relação a distribuição esperada, como mostrado nos histogramas da Figura 21, como esses gráficos buscava-se compreender quão eficiente os modelos são em captar a variação real dos dados.

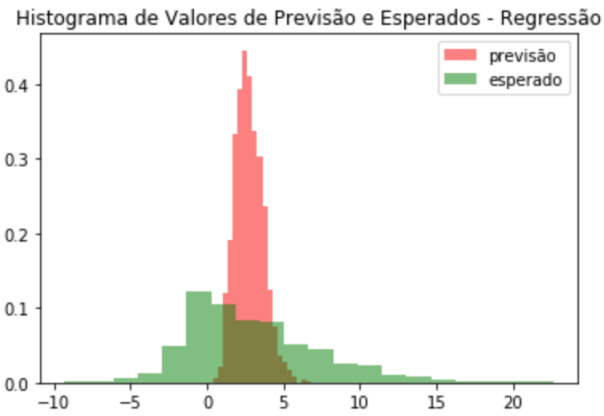
Figura 20: Métricas de predição do modelo para os melhores times previstos

model_name	pred_avg	pred_dev	desired_avg	desired_dev	mean_abs_error	root_mean_sqrt_error	r2_score
knn	3.669477	0.140966	4.293452	2.332469	3.425459	3.838925	-48.768815
xgb	4.829164	0.316827	4.461905	3.167095	3.883526	4.353982	-11.432090
linear	4.178988	0.207392	3.614286	2.823056	3.658717	4.078679	-34.845903

Fonte: Próprio Autor.

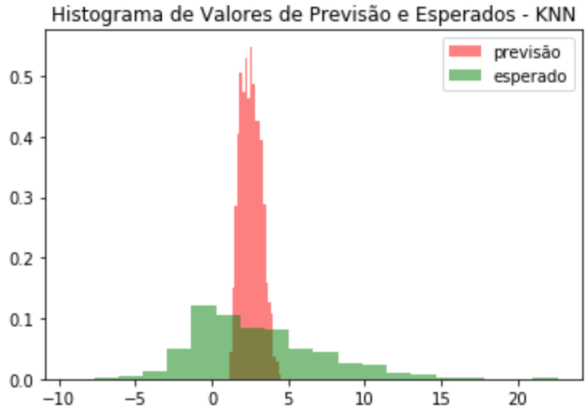
Figura 21: Histograma de Distribuição da Pontuação

(a) Regressão



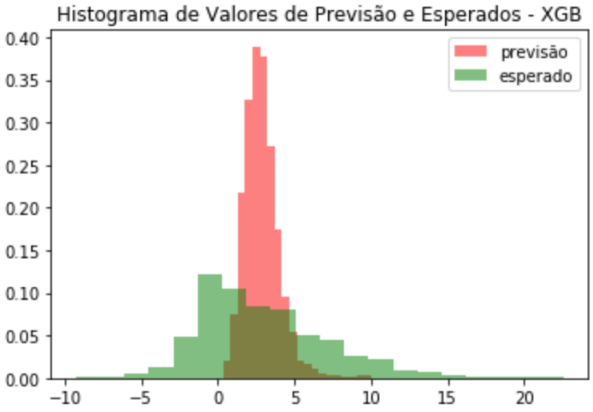
Fonte: Próprio Autor.

(b) KNN



Fonte: Próprio Autor.

(c) XGBoost



Fonte: Próprio Autor.

5 DISCUSSÃO

Nesse capítulo serão analisados os resultados obtidos na seção anterior, buscando compreender e explicar as informações que podem ser extraídas a partir desses resultados. Além disso, será realizado um estudo comparativo entre os diferentes modelos para tentar concluir as principais vantagens e desvantagens de cada um, e decidir, para a aplicação em questão qual deles deveria ser utilizado.

5.1 Métricas de Execução

A partir da análise dos resultados de execução é possível observar que mesmo para aplicações cuja quantidade de dados não é tão grande, a quantidade de armazenamento e o tempo necessário para execução dos algoritmos é razoavelmente elevada. Esta dificuldade, somada a outros fatores, como aumento do potencial computacional, diminuição dos custos relacionados ao armazenamento e utilização de dados, dentre outros elementos, parece explicar o o crescimento exponencial de *machine learning* nos últimos anos, dado o crescimento do poder computacional e a diminuição dos custos relacionados a armazenamento e utilização de dados.

Pode-se observar também que a estrutura de bancos relacionais muitas vezes torna-se lenta em aplicações que exigem pesquisas muito recorrentes, como o cálculo de *features* em uma aplicação de *machine learning*. Para contornar essa dificuldade, é muito comum a utilização de computação distribuída, que possibilita a realização de cálculos em memória, resultando em uma diminuição significativa do tempo de execução, sobretudo em se tratando de situações, comuns hoje, em que a quantidade de dados é grande.

5.2 Análise dos dados

A análise dos dados a serem utilizados na modelagem é uma etapa fundamental, já que é a partir dela que surgem as ideias e visualizações necessárias para compreender

como os dados existentes influenciam o resultado desejado e cujo modelo tentará prever. O resultado analítico demonstrou que a maioria das "crenças"existentes sobre o futebol podem ser comprovadas com dados. Por exemplo, a dominância dos times que jogam em casa é algo muito dito por comentaristas esportivos, e partir dos dados foi possível observar que, de fato, por exemplo, o time visitante só vence a partida em 23.2% dos jogos, o que demonstra que realmente, o fator "casa"é bastante determinante no resultado. Ainda analisando esse fator, observou-se como o desempenho dos jogadores varia jogando dentro ou fora de casa e viu-se que para cada posição esse efeito é diferente, mas que para todas, o desempenho fora de casa é inferior ao desempenho dentro de casa.

Realizou-se também um estudo da desempenho dos times e viu-se que como esperado a desempenho dos times fora de casa também é pior do que aquela que apresentam dentro de casa. No entanto, observa-se que os times que em geral se destacam nas competições nacionais, tem uma desempenho fora de casa bastante superior a dos demais times, o que mostra um outro fator a ser considerado, a importância de ser um time forte fora de casa.

Essa análise foi bastante importante para criação das *features*. Percebeu-se a importância de ter *features* que distinguam bem os dados dentro e fora de casa, e, por isso, para a maioria das *features*, foi calculado o valor dela tanto dentro quanto fora de casa. Por exemplo, com relação à característica Gols Feitos nos Últimos 5 Jogos foi divida em Gols Feitos nos Últimos 5 Jogos em Casa e Gols Feitos nos Últimos 5 Jogos Fora de Casa. Procedendo assim o que o modelo é capaz de aprender a importância de ser o time anfitrião ou não.

A segunda etapa da análise teve como objetivo compreender como algumas principais características estão relacionadas ao próprio desempenho dos jogadores, para isto analisamos como a desempenho do time (número de vitórias, empates e derrotas) e a média de gols do time, se relacionam como a pontuação dos jogadores. Observando os gráficos é possível ver que existe uma relação direta entre essas variáveis e que portanto estas características serão fundamentais para previsão do desempenho dos jogadores.

Para finalizar, realizou-se uma análise por posição para entender como a desempenho dos jogadores tem comportamento diferente dependendo da posição do jogador. Inicialmente, observou-se que em média goleiros e laterais são as posições com melhor média. No entanto, se observarmos apenas as 10% melhores pontuações de cada posição, observa-se que goleiros e atacantes são os com a maior média de pontuação. Isso demonstra a importância de escolher bons atacantes, já que em média são jogadores que pontuam pouco, no entanto, os melhores atacantes estão entre os jogadores que mais pontuam. A seguir,

para compreender quais são as jogadas que mais afetam a pontuação dos jogadores, foram analisados os gráficos das Figuras 24 à 28, em que se observaram alguns fatos bastante interessantes. A primeira observação interessante é que "Roubadas de Bola"são uma das jogadas mais determinantes na pontuação de jogadores para a maioria das posições, isto porque no jogo Cartola, a "roubada de bola"vale 2,5 pontos, o que representa uma pontuação bastante elevada. Além disso, é possível observar que para os zagueiros e laterais não tomar gols é uma das coisas que mais contribui para sua pontuação enquanto que para meias e atacantes, fazer gols é fundamental.

5.3 Análise dos Modelos

Nessa seção avaliaremos os resultados do treinamento dos três modelos utilizados para compreender como cada métrica reflete a desempenho do modelo e como pode-se utilizar essas métricas para avaliar qual o modelo mais recomendado para a situação.

Vale lembrar que o objetivo final do modelo é o de otimizar a pontuação obtida em cada rodada pela formação de um time de 11 jogadores, respeitando as restrições de posição. Por isso, o melhor modelo é aquele que maximiza a média de pontuação ao realizar essa escolha de 11 jogadores. No entanto, existem outras medições que são bastante úteis para compreender se o modelo é uma boa representação para os dados apresentados. Por exemplo, a métrica R-Quadrado mede como a variância real dos dados está representada na variância prevista dos dados, isto é, quão representativo o modelo é.

Observando os dados obtidos vemos que o modelo que maximiza a média de pontuação de cada escalação é o *XGBoost* no qual pontuação média dos jogadores escolhidos é de 4.461905 como podemos ver na Figura 20 e portanto podemos dizer que este é o melhor modelo para a aplicação em questão. A partir desse valor, pode-se determinar que o modelo teria uma média de aproximadamente 54 pontos por rodada, o que representa uma pontuação bastante competitiva em relação a pontuação média dos participantes do jogo.

No entanto, quando observamos a métrica R-Quadrado e os histogramas da distribuição da predição em comparação com a distribuição real da pontuação, é possível observar que nenhum dos modelos consegue captar a variância real dos jogadores, e tende fazer uma previsão próxima a média dos jogadores. Isto é, a capacidade de compreender os fatores que levam um jogador a se destacar não são bem compreendidas pelo modelo.

Muitas são as explicações plausíveis para esta observação. A primeira parece ser uma

decorrência do fato de que a distribuição da pontuação dos jogadores é muito concentrada nos valores médios, isto é, existem poucos dados que possuem pontuação muito alta ou muito baixa, e portanto o modelo não consegue captar esse comportamento. O segundo ponto, é que os dados coletados a partir do Cartola não refletem muitas das características dos jogadores e dos times que podem ser decisivas para prever a pontuação dos jogadores, como a posse de bola, ou o deslocamento na partida.

No entanto, mesmo sem que nenhum modelo capte de forma completa o comportamento dos jogadores a partir dos dados, o modelo que utiliza o algoritmo *XGBoost* é o que apresenta o melhor resultado, tanto quantitativamente em relação à métrica de pontuação média nas rodadas testadas, quanto em relação a capacidade de compreender o comportamento dos jogadores que se destacam, como mostrado no histograma 21 c), em comparação aos histogramas 21 a) e 21 b).

6 CONCLUSÃO

Nessa seção serão revistas as informações obtidas durante o desenvolvimento do projeto. Então será realizada uma análise sobre as contribuições do projeto para as área afins. Finalmente, serão levantados os pontos de melhoria e os próximos passos necessários para dar continuidade ao projeto.

6.1 Aprendizados Adquiridos

O projeto foi considerado bastante satisfatório em termos didáticos já que a partir dele foi possível obter diversos aprendizados sobre diversas áreas de conhecimento. Primeiramente, foi possível expandir os conhecimentos acerca das ferramentas existentes para criação de bancos de dados, e compreender as vantagens e desavantagens de cada um. Foi possível também entender como ocorre o processo de criação de *features* e como o conhecimento da área de aplicação é fundamental para criação de características relevantes. Então, na etapa de modelagem, foi possível observar com as etapas de pré processamento e escolha de parâmetros são fundamentais para criação de um modelo com boa desempenho. Além disso, foi possível compreender os prós e contras de cada tipo de modelo e como as especificidades de cada método devem ser levadas em conta para a escolha de um algoritmo adequado para o problema em questão. A seguir, observou-se como a escolha e a análise cuidadosa das métricas de desempenho pode ser fundamental para entender qual o melhor modelo e qual a desempenho esperada dele quando ele for utilizado em uma aplicação real.

6.2 Contribuições do Projeto

Estudando as diversas etapas do projeto é possível levantar diversas contribuições do projeto tanto para a área de *machine learning* quanto para a área esportiva. A principal contribuição do projeto, acredita-se, é justamente a integração entre a área esportiva e

machine learning já que essa é uma aplicação de aprendizado de máquina muito pouco explorado, e para a qual o projeto teve o intuito de demonstrar a sua aplicabilidade. Para a área esportiva, o projeto apresenta-se como uma análise de dados bastante interessante e que demonstra de forma quantitativa, diversas características que podem ser interessantes para especialistas da área. Da mesma maneira, o projeto faz uma análise concreta da aplicação de diversos algoritmos de modelagem, o que pode ser utilizado como referência dos prós e contras de cada método para futuras aplicações de *machine learning* na área esportiva ou em outras áreas. Considerando a aplicação prática do algoritmo para jogar o jogo Cartola, pode-se concluir que segundo os testes realizados a média de pontos do algoritmo seria de aproximadamente 50 pontos, o que representa uma pontuação acima da média dos demais participantes no jogo Cartola. Um aspecto não considerado neste desempenho foi a questão econômica, muitas vezes determinante na escolha dos jogadores e que poderia comprometer um pouco a desempenho do modelo. Ainda assim, o sistema se mostrou viável para realização de uma boa escalação no Cartola.

6.3 Próximos Passos

Após uma análise cuidadosa da metodologia realizada, bem como dos resultados obtidos, foram listados diversos pontos de melhoria e as possíveis ações necessárias para gerar essas melhorias.

6.3.1 Coleta de Dados

O aumento da quantidade de dados é fundamental para melhorar a desempenho do modelo. A quantidade de dados aumenta naturalmente como o tempo já que o campeonato continua acontecendo e assim mais dados são coletados. No entanto, como o número de jogos por rodada é bastante limitado, a quantidade de dados cresce lentamente. Portanto, para aumentar mais rapidamente a quantidade de dados, seria necessário buscar outras fontes de dados como outros campeonatos ou torneios, o que, como consequência, traria um aumento na quantidade de dados em detrimento da uniformidade das informações.

Outro ponto interessante em relação a coleta de dados, seria a de buscar fontes de informação com dados adicionais sobre os dados já existentes. Seria possível, por exemplo, obter dados sobre posse de bola ou sobre jogadores lesionados, e estes dados, em cada rodada, poderiam melhorar as informações passadas ao modelo.

6.3.2 Treinamento

Além de aumentar a quantidade de dados, é possível melhorar a desempenho do modelo de diversas outras maneiras. Com esse objetivo, a continuidade do projeto deveria se preocupar com o vários elementos relacionados a modelagem como o preenchimento dos valores nulos, a normalização dos dados e a escolha dos parâmetros do modelo, cujo valor ótimo varia com as amostras presentes.

6.3.3 Utilização

O principal passo na continuidade do projeto é conseguir utiliza-lo para jogar de forma automatizada, o que requer garantir que o modelo seja estável para dados futuros. Além disso, deve-se adicionar ao algoritmo as restrições econômicas presentes no jogo, que poderiam limitar a desempenho do modelo, ao restringir os jogadores que podem ser escalados. Não obstante, poderia-se criar uma infraestrutura que permitisse a coleta automática de dados a cada rodada, e que esses novos dados já fossem utilizados para retreinamento do modelo, adicionando bastante robustez ao sistema. Finalmente, esse novo modelo poderia ser utilizado para avaliar os jogadores de uma rodada e acessar a API do Cartola para escalação do time.

REFERÊNCIAS

1 SZEGEDY, C. et al. Going deeper with convolutions. In: . IEEE, 2015. p. 1–9. ISBN 978-1-4673-6964-0. Disponível em: ⟨http://ieeexplore.ieee.org/document/7298594/⟩.

2 KALLENBERG, M. et al. Unsupervised Deep Learning Applied to Breast Density Segmentation and Mammographic Risk Scoring. *IEEE Transactions on Medical Imaging*, v. 35, n. 5, p. 1322–1331, maio 2016. ISSN 0278-0062, 1558-254X. Disponível em: ⟨http://ieeexplore.ieee.org/document/7412749/⟩.

3 MADGE, S. Predicting Stock Price Direction using Support Vector Machines. p. 14.

4 MAGALHÃES, M. N. *Noções de Probabilidade e Estatística*. 6. ed. [S.l.]: EDUSP, 2004. v. 1.

5 Blog, REGRESSION Analysis: How Do I Interpret R-squared and Assess the Goodness-of-Fit? 2013. Disponível em: ⟨http://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit⟩.

6 HASTIE, T.; Tibshirani, R. ; Friedman, H. *The Elements Of Statistical Learning*. 2. ed. [S.l.]: Springer, 2017. v. 1.

7 Tavish SRIVASTAVA. *Introduction to k-Nearest Neighbors: Simplified*. 2018. Disponível em: ⟨https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/⟩.

8 INTRODUCTION to Boosted Trees. 2015. Disponível em: ⟨https://xgboost.readthedocs.io/en/latest/model.html⟩.

9 JI, B.; LI, J. NBA All-Star Lineup Prediction Based on Neural Networks. In: . IEEE, 2013. p. 864–869. ISBN 978-1-4799-5245-8. Disponível em: ⟨http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6973701⟩.

10 HUCALJUK, J.; RAKIPOVIć, A. Predicting football scores using machine learning techniques. p. 5.

11 Tianxiang Cui et al. An ensemble based Genetic Programming system to predict English football premier league games. In: . IEEE, 2013. p. 138–143. ISBN 978-1-4673-5855-2. Disponível em: ⟨http://ieeexplore.ieee.org/document/6604116/⟩.

12 HUANG, K.-Y.; CHANG, W.-L. A neural network method for prediction of 2006 World Cup Football Game. In: . IEEE, 2010. p. 1–8. ISBN 978-1-4244-6916-1. Disponível em: ⟨http://ieeexplore.ieee.org/document/5596458/⟩.

13 GOMIDE, H. *caRtola: Extração de dados da API do CartolaFC, análise exploratória dos dados e modelos preditivos em R e Python - 2014-17. [EN] Data munging, analysis and modeling of CartolaFC - the most popula.*. 2018. Original-date: 2016-05-24T20:21:32Z. Disponível em: ⟨https://github.com/henriquepgomide/caRtola⟩.

APÊNDICE A – CÓDIGO FONTE

Todo o código fonte pode ser encontrado em ⟨https://github.com/NoixD/cartolaAPI⟩. O código para geração das características está em ⟨https://github.com/NoixD/cartolaAPI/tree/master/notebooks/FeatureCalculation.ipynb⟩. O código de treinamento está em ⟨https://github.com/NoixD/cartolaAPI/tree/master/notebooks/Training.ipynb⟩. O código para avaliação de desempenho do modelo está em ⟨https://github.com/NoixD/cartolaAPI/tree/master/notebooks/ModelEvaluation.ipynb⟩.