UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
UNIDADE ACADÊMICA DE SISTEMAS E COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LUCAS BARROS ROCHA

VENCENDO O JOGO

CAMPINA GRANDE - PB

2019

LUCAS BARROS ROCHA

VENCENDO O JOGO

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Professor Dr. Leandro Balby Marinho.

CAMPINA GRANDE - PB

2019

R672v Rocha, Lucas Barros.
Vencendo o jogo. / Lucas Barros Rocha. – 2019.

13 f.

Orientador: Prof. Dr. Leandro Balby Marinho.
Trabalho de Conclusão de Curso - Artigo (Curso de Bacharelado em Ciência da Computação) - Universidade Federal de Campina Grande; Centro de Engenharia Elétrica e Informática.

1. Predição de resultados - futebol. 2. Futebol – predição de resultados. 3. Apostas. 4. Casas de apostas. 5. Algoritmos preditivos. 6. Gradient Boosting. I. Marinho, Leandro Balby. II. Título.

CDU:004(045)

Elaboração da Ficha Catalográfica:

Johnny Rodrigues Barbosa
Bibliotecário-Documentalista
CRB-15/626

LUCAS BARROS ROCHA

VENCENDO O JOGO

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Professor Dr. Leandro Balby Marinho
Orientador – UASC/CEEI/UFCG

Professor Dr. Cláudio Elízio Calazans Campelo
Examinador – UASC/CEEI/UFCG

Professor Dr. Tiago Lima Massoni
Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 02 de julho de 2019.

CAMPINA GRANDE - PB

Vencendo o Jogo

Lucas Barros Rocha
UFCG - Universidade Federal de Campina Grande
Campina Grande, Paraíba
lucas.rocha@ccc.ufcg.edu.br

RESUMO

Com a quantidade de dados disponibilizados na Internet nos últimos anos relacionados ao futebol, surgiu o interesse de conhecer mais sobre o esporte e tentar prever resultados de partidas a partir do conhecimento adquirido com resultados históricos. Este trabalho propõe um método para predição de resultados de partidas de futebol utilizando mineração de dados. Para tal, foram feitas extrações de dados a partir de portais online e sites de casas de apostas que oferecem dados publicamente, para formação de uma base de dados integrada. Em seguida, foi aplicado um processo de limpeza, organização e derivação de dados, para que algoritmos de aprendizagem de máquina pudessem buscar padrões implícitos nos dados, com o intuito de obter o melhor valor de predição. Foram alcançados acertos superiores aos das casas de apostas as quais, por si só, podem ser consideradas boas preditoras. Na predição de vitória do time da casa, por exemplo, foi obtido um valor de predição 12,14 % superior ao das casas de apostas.

KEYWORDS

futebol, apostas, algoritmos, predição, casas de apostas

1 INTRODUÇÃO

Os esportes sempre foram uma atividade que chama a atenção de todos, tanto como entretenimento quanto como atividade de cultura e lazer. Dentre as diversas modalidades, pode-se destacar o futebol, considerado o esporte mais popular do mundo. Segundo a FIFA – Federação Internacional de Futebol Association – são mais de 250 milhões de praticantes, dos quais 40 milhões são profissionais distribuídos em quase 300 mil clubes de 207 países [8].

Essa popularização do esporte aliada à democratização da Internet impulsionaram o crescimento dos mercados de apostas pelo mundo na última década. No Brasil, as apostas esportivas não são regulamentadas, porém existem inúmeros sites de outros países que operam na língua portuguesa e aceitam pagamentos por meio de boletos e transferências bancárias.

Nos últimos anos, com a evolução das técnicas de aquisição e armazenamento de dados, uma grande quantidade de informações passou a ser disponibilizada na Internet relacionados às partidas de futebol disputadas, escalações, resultados, estatísticas e cotações de mercados de apostas.

Esse projeto foca na construção de modelos que usem mineração de dados para realizar previsões de resultados de futebol e possam auxiliar os apostadores na tomada de decisão. Sendo assim, este estudo se caracteriza tecnicamente como um processo interativo de extração de conhecimento em bases de dados (*KDD - Knowledge Discovery in Databases*) [7], focado em identificar novos padrões implícitos.

Também buscou-se com este estudo criar um modelo para predição de resultados de partidas de futebol, utilizando mineração de dados com algoritmos de aprendizagem de máquina. Para tal, teve-se como objetivos específicos: criar, transformar e selecionar atributos relevantes para predição de resultados (*feature engineering*); comparar diferentes abordagens para seleção de atributos; e comparar diferentes técnicas de aprendizagem de máquina para serem aplicadas no modelo de predição de resultados.

A pergunta de pesquisa que este trabalho pretende responder é: *"técnicas de aprendizagem de máquina que utilizam dados históricos do mercado de apostas conseguem superar o próprio mercado em termos de predição de resultados de partidas?"*.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta a fundamentação teórica necessária para o entendimento deste trabalho. Primeiramente, são apresentadas informações sobre o mercado de apostas esportivas e, em seguida, são apresentados os algoritmos de aprendizagem de máquinas utilizados neste trabalho.

As casas de apostas são empresas que aceitam apostas dos clientes (apostadores) na previsão de um certo acontecimento, com um potencial de lucro calculado com base nas probabilidades da ocorrência destes eventos. Atualmente, as casas de apostas atuam principalmente de forma online, permitindo realizar apostas em jogos esportivos, a partir de qualquer lugar do mundo, restringindo apenas países que proíbem em sua legislação as apostas/jogos de azar.

Nestes estabelecimentos, em um jogo disponível para apostas, existem diversos mercados em que um apostador pode atuar. Em uma partida de futebol, por exemplo, existem os seguintes mercados: mercado de gols, mercado de cartões, mercado de escanteios, entre outros. Outro exemplo é o mercado do resultado exato do final da partida. Nele, encontra-se o maior número de opções (seleções): 0-0, 1-0, 0-1, 2-0, 2-1, 0-2, e assim por diante.

Entretanto, o principal deles é o mercado *Money Line*, conhecido também como *Match Odds*. Este mercado refere-se ao resultado final da partida, existindo três seleções de mercado: vitória do time da casa, empate e vitória do time visitante. As seleções são todas as possíveis opções que um apostador pode escolher. São nas seleções que estão presentes as *odd*.

Uma odd corresponde à probabilidade de um determinado evento ocorrer em um mercado. Seu cálculo é feito a partir da análise estatística feita pelas casas de apostas que aponta a perspectiva daquele duelo. Geralmente, é representada por um número real, maior que um.

Para entender melhor as *odd*, considere o seguinte cenário: um jogo entre Corinthians e Atlético-PR (ver Figura 1). O Corinthians possui *odd* de 1.83 para vencer o Atlético-PR. Isso significa que, a cada R$ 1,00 apostado, haverá um retorno ao apostador de R$ 1,83 (R$ 1,00 que é o valor da aposta + R$ 0,83 de lucro).

Para saber a real probabilidade de vitória do Corinthians, é necessário realizar o seguinte cálculo: dividir o número 1 pelo valor da odd, que é 1.83 (ver Figura 1). Em seguida, deve-se multiplicar o resultado por 100, e assim encontra-se a porcentagem de chance de vitória do Corinthians, que é igual a 54,64%. Vejamos:

$$1/1.83 = 0{,}5464 \times 100 = 54{,}64\%$$

	B's	1	X	2
Botafogo RJ - Santos	23	2.52	3.08	2.86
Gremio - Flamengo RJ	23	2.38	2.94	3.18
Corinthians - Atletico-PR	23	1.83	3.35	4.38
America MG - Palmeiras	23	3.15	3.19	2.26
Parana - Ceara	23	2.05	3.04	3.89
Sao Paulo - Vasco	23	1.38	4.29	8.32
Vitoria - Cruzeiro	23	2.57	3.21	2.65
Fluminense - Bahia	21	2.00	3.24	3.81
Sport Recife - Chapecoense-SC	23	1.96	3.18	3.98
Atletico-MG - Internacional	16	2.02	3.26	3.63
Ceara - Santos				

Figura 1: Jogos da rodada 23 do campeonato brasileiro do ano de 2018, via BetExplorer. (https://www.betexplorer.com/soccer/brazil/serie-a-2018/)

Um detalhe importante sobre as *odd* é que, somando as probabilidades das seleções de um mercado, o resultado sempre ultrapassa os 100%, pois se a soma das probabilidades fosse menor que isto, poder-se-ia apostar em todas as seleções do mercado e ainda assim obter lucro. Voltando ao jogo do Corinthians contra Atlético-PR (ver Figura 1), pode-se entender esta situação:

- Probabilidade do Corinthians vencer a partida: 1/1.83 = 0,5464 x 100 = 54,64%
- Probabilidade de ocorrer o empate: 1/3.35 = 0,2985 x 100 = 29,85%
- Probabilidade do Atlético-PR vencer a partida: 1/4.38 = 0,2283 x 100 = 22,83%

Percebe-se que a soma das três probabilidades é igual a 107,32%. Neste cenário, pode-se afirmar que a casa de apostas obterá, no mínimo, 7,32% de lucro caso o cliente (apostador) aposte nas 3 seleções possíveis.

Em relação aos algoritmos, foram utilizados os seguintes neste trabalho:

Algoritmos de aprendizagem de máquina

- *Random Forest*: é um algoritmo de aprendizagem supervisionada que cria uma floresta de possibilidades aleatórias. A "floresta" é uma combinação (ensemble) de árvores de decisão, na maioria dos casos treinados com o método de *bagging*. A ideia principal do método de *bagging* é que a combinação dos modelos de aprendizado aumenta o resultado geral;
- *Gradient Boosting*: é uma técnica de aprendizado de máquina para problemas de regressão e classificação que produz um modelo de previsão na forma de um conjunto de modelos de previsão fracos, geralmente árvores de decisão [6];
- *AdaBoost*: é um algoritmo de aprendizado supervisionado do tipo boost. O *AdaBoost* combina um conjunto de funções simples de classificação, denominadas classificadores fracos, para formar um classificador forte.

Modelo de regressão linear

- *Logistic Regression*: é uma técnica estatística que tem como objetivo produzir, a partir de um conjunto de informações, um modelo que permita a predição de valores tomados por uma variável categórica, frequentemente binária, a partir de uma série de variáveis explicativas contínuas e/ou binárias.

3 METODOLOGIA

Para alcançar os objetivos desta pesquisa, fez-se necessária a utilização de uma base de dados integrada e estruturada com informações relacionadas às partidas de futebol como, por exemplo, resultados e estatísticas de partidas, cotações dos mercados de apostas, etc.

Como não existe uma base de dados pública com estes dados, foi necessária a extração de informações de sites que disponibilizam as informações publicamente, tais quais: *Bet Explorer* [4], o qual oferece dados sobre resultados de partidas, classificação de campeonatos e *odd* de diferentes casas de apostas; *BetFair* [5], que oferece dados sobre *odd* de partidas de futebol; e *Football Data* [9], que oferece dados sobre partidas de futebol, cotações de diversas casas de apostas e

detalhes técnicos das partidas como, por exemplo, nome do juiz, horário e escalação dos times.

A extração das informações para criação da base de dados foi realizada em duas etapas. A primeira consistiu no desenvolvimento de *web crawlers* que realizam o download de páginas web contendo informações das partidas desejadas. Entretanto, como os dados dos arquivos baixados não são estruturados, foi necessário realizar uma segunda etapa chamada de *web scraping*, na qual é feita uma varredura nos arquivos, extraindo as informações desejadas e inserindo-as em uma base de dados estruturada. Esses procedimentos foram realizados para cada um dos três sites supracitados, formando uma base centralizada e integrada.

Após a consolidação da base de dados, foi iniciado o processo de descoberta de conhecimento (do inglês, *Knowledge Discovery in Databases - KDD*) para que pudessem ser realizadas as predições de resultados. O processo foi dividido nas seguintes etapas [10]:

- Seleção dos atributos relevantes na base de dados (feature engineering);
- Transformação dos dados de forma que seja facilitado o uso de técnicas de mineração de dados;
- Criação de um modelo para predição de resultados usando algoritmos para descoberta de padrões (mineração de dados e aprendizagem de máquina);
- Interpretação e avaliação dos resultados para verificar se o modelo proposto é válido.

Preparado o ambiente de trabalho, foi escolhida a linguagem de programação *Python* v3.6.4 [15] para implementação e o *framework Anaconda* v3.6.3 [2] que possibilita organizar o ambiente virtual, gerenciar pacotes e bibliotecas e desenvolver estudos baseados em ciência de dados e aprendizagem de máquina. Para gerenciar a base de dados, foi utilizado o SGBD *MySQL* v4.0 [11], por ser gratuito e robusto, além de oferecer suporte multiplataforma.

Uma vez preparado o ambiente de trabalho, foram selecionados os portais online que oferecem dados confiáveis e relevantes para o estudo relacionados às estatísticas, cotações e condições de partidas de futebol. Para extrair dados desses portais, foi utilizada a biblioteca *Requests* [16]. Com o seu uso, foi possível escrever algoritmos de *web crawler*, os quais permitem realizar o download das páginas web na forma de arquivos contendo dados estruturados. Como estas informações ainda não estavam organizadas e formatadas para serem inseridas na base de dados *MySQL*, foram criados algoritmos de *web scraper* utilizando a biblioteca *Beautiful Soup* [3], que permite acessar os arquivos baixados e extrair as informações desejadas de forma estruturada.

Após a extração e formatação dos dados, os mesmos foram armazenados no banco de dados MySQL com auxílio da biblioteca *SQLAlchemy* [18]. Tal biblioteca permite salvar e acessar informações no banco de dados, prezando pela segurança, consistência e organização dos dados.

As informações de cada fonte foram integradas e armazenadas na base de dados centralizada de forma que pudessem ser consultadas mediante a realização de consultas SQL. Por exemplo, uma consulta sobre as informações de um determinado evento (jogo) ou campeonato.

Após a criação da base de dados consolidada, com o uso da biblioteca *SQLAlchemy*, foi extraído um *data frame*, ou seja, um arquivo cuja estrutura interna é semelhante a uma matriz na qual as colunas possuem nomes e podem conter dados de tipos diferentes. O *data frame* pode ser visto como uma tabela da base de dados, em que cada linha corresponde a um registro da tabela. O *data frame* é necessário para possibilitar a análise exploratória dos dados, manipulação e organização até que os mesmos possam ser utilizados pelos algoritmos de aprendizagem de máquina.

O *data frame* inicialmente gerado possui um total de 37.540 jogos referentes a nove campeonatos de futebol: Brasil (*Série A*), Brasil (*Série B*), Inglaterra (*Premier League*), Espanha (*La Liga*), Alemanha (*Bundesliga*), Itália (*Serie A*), França (*Ligue 1*), Portugal (*Primeira Liga*) e Holanda (*Eredivisie*). Os jogos são de 12 temporadas, de 2006 a 2017. As temporadas dos campeonatos do Brasil (*Séries A e B*) começam e terminam no mesmo ano. As temporadas dos campeonatos europeus tem início no segundo semestre do ano e terminam no primeiro semestre do ano seguinte. Para padronizar, decidiu-se referenciar as temporadas dos campeonatos europeus pelo ano em que começam. Por exemplo, a temporada 2006/07 foi identificada como a temporada 2006, semelhante às temporadas que acontecem no Brasil.

A Figura 2 apresenta um resumo dos jogos por campeonato e temporada, contidos no *data frame* gerado. Nela, é possível observar três detalhes importantes: a) alguns campeonatos tem menos jogos do que outros, pois a quantidade de times é menor; b) o campeonato de Portugal a partir do ano 2014 passou de 16 para 18 times e teve a quantidade de jogos aumentada; c) quando o *data frame* foi gerado, a temporada 2017 dos campeonatos europeus não tinha sido concluída, apresentando portanto um número menor de partidas. A última coluna exibe quantidade de partidas por campeonato enquanto a última linha mostra a quantidade de partidas por temporada.

Antes de começar a manipular os dados, é necessário observar como os mesmos estão estruturados e organizados, visando detectar erros ou falhas nos dados de maneira que possam ser corrigidos ou desconsiderados. Para observar a estrutura dos dados, foram selecionados alguns dos atributos das partidas: Nome do Campeonato, Ano da Temporada, Time da Casa, Time Visitante, Gols Marcados pelo Time da Casa, Gols Marcados pelo Time Visitante e Time Vencedor

Ano	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	Total
Campeonato													
Portugal A	240	240	240	240	240	240	240	240	306	306	306	270	3108
Germany A	306	306	306	306	306	306	306	306	306	306	306	270	3636
Netherlands A	306	306	306	306	306	306	306	306	306	306	306	281	3647
Italy A	380	380	380	380	380	380	380	380	380	380	380	321	4501
Spain A	380	380	380	380	380	380	380	380	380	380	380	323	4503
France A	380	380	380	380	380	380	380	380	380	380	380	330	4510
England A	380	380	380	380	380	380	380	380	380	380	380	335	4515
Brazil A	380	380	380	380	380	380	380	380	380	380	380	380	4560
Brazil B	380	380	380	380	380	380	380	380	380	380	380	380	4560
Total	3132	3132	3132	3132	3132	3132	3132	3132	3198	3198	3198	2890	37540

Figura 2: Resumo dos dados referente a quantidade de partidas por temporada e campeonato. Elaboração própria.

da Partida. Os valores que cada atributo pode assumir são intuitivos, exceto no que se refere ao último, que pode assumir os valores h (*HOME* - time da casa), d (*DRAW* - empate) ou a (*AWAY* - time visitante), como pode ser observado na Figura 3.

	Campeonato	Ano	T. Casa	T. Visitante	GC	GV	Resultado
8314	Brazil A	2016	Chapecoense-SC	Flamengo RJ	1.0	3.0	a
1565	Brazil A	2010	Flamengo RJ	Gremio	1.0	1.0	d
22327	Germany A	2013	Augsburg	B. Monchengladbach	2.0	2.0	d
32246	Netherlands A	2014	AZ Alkmaar	Heerenveen	0.0	1.0	a
36806	Brazil A	2017	Flamengo RJ	Vitoria	0.0	2.0	a
23193	Portugal A	2012	Guimaraes	Braga	0.0	2.0	a
14114	Italy A	2007	Fiorentina	Livorno	1.0	0.0	h
16951	Germany A	2011	Hoffenheim	B. Monchengladbach	1.0	0.0	h
30112	Portugal A	2015	Arouca	Maritimo	4.0	1.0	h
10790	Brazil B	2014	Boa	Joinville	1.0	0.0	h

Figura 3: Exibição de dados selecionados a partir do *data frame*. Elaboração própria.

Verificou-se nos dados a existência de valores não preenchidos (*missings*), especificamente, se existia alguma partida que não possuísse o valor preenchido no atributo "resultado da partida". Observando a Figura 4, percebe-se que foi encontrado apenas um jogo sem o resultado da partida: Chapecoense-SC x Atlético-MG, pelo Campeonato Brasileiro da Série A de 2016. Isso ocorreu devido ao acidente aéreo com a equipe da Chapecoense e a consequente não realização de algumas partidas daquele campeonato. Com isso, esta partida foi removida do *data frame*, pois não poderia ser usada para prever o resultado de outras partidas.

As *odd* das casas de apostas constituem um atributo importante para predição de resultados de partidas. A partir do valor definido para as *odd*, pode-se chegar a algumas conclusões como, por exemplo, se a partida possui um favorito, se a partida tem tendência a ocorrer muitos/poucos

	Campeonato	Ano	T. Casa	T. Visitante	GC	GV	Resultado
7790	Brazil A	2016	Chapecoense-SC	Atletico-MG	NaN	NaN	NaN

Figura 4: Jogos sem o resultado final da partida. Elaboração própria.

gols/escanteios, entre outras. Por esta razão, o presente trabalho fez uso de *odd* como atributos (*features*) importantes para a concepção de um modelo preditivo de resultados de partidas.

Verificou-se, também, se existem *odd* associadas a todas as partidas. Percebeu-se que nem todas as partidas contidas no *data frame* contém *odd*, principalmente as que aconteceram nos anos 2006 e 2007. Foram, então, desconsideradas as partidas que não possuem os valores de *odd* correspondentes ao mercado *Money Line*. Com isso, dos 37.540 jogos iniciais, restaram 36.388 jogos disponíveis para serem trabalhados. Com isto, conclui-se a primeira etapa do trabalho, que era a construção do banco de dados integrado.

Passou-se para a segunda etapa do *KDD*, que consistia na transformação dos dados de forma a facilitar a aplicação das técnicas de mineração. Neste sentido, a partir do valor das *odd*, foram derivados novos atributos utilizando a biblioteca Pandas [12]. Tal biblioteca permite manipular o conteúdo do *data frame*, criar novos atributos a partir da seleção de outros já existentes e salvar tudo em um novo *data frame* ou mesmo substituir o *data frame* existente.

Os atributos derivados a partir das *odd* referentes ao mercado *Money Line* foram: *favorite*, indica qual dos três resultados (vitória do time da casa, empate ou vitória do time de fora) é o favorito segundo as casas de apostas; *underdog* ("azarão"), indica a opção com a menor probabilidade de ser o resultado final da partida; e *medium*, que assume como valor a seleção que tem a probabilidade menor do que o *favorite* e maior do que a do *underdog*.

Como esclarecido na seção de Fundamentação Teórica, nos mercados, a soma do valor das *odd* de uma partida sempre ultrapassa 100%. Isto porque as *odd* são constituídas da probabilidade real de um evento ocorrer e da margem de lucro da casa de apostas. No entanto, para o presente estudo, é necessário considerar apenas a probabilidade real de cada resultado possível em uma partida (vitória do time da casa, empate, vitória do time de fora), pois o algoritmo elaborado não trabalha com esta margem de lucro. Desta forma, foram derivados outros três atributos: probabilidade real do time casa vencer a partida, probabilidade real do empate ser o resultado final da partida e probabilidade real do time de fora vencer a partida.

Para calcular a probabilidade real de uma seleção, é necessário transformar a *odd* da seleção em seu inverso e, em seguida, dividir este resultado pelo somatório do inverso das

demais *odd* da mesma seleção. O n da fórmula é o número de seleções do mercado, como mostra a Equação 1.

$$\frac{\frac{1}{oddx}}{\sum_{k=1}^{n} \frac{1}{oddk}} \quad (1)$$

Observando os atributos das probabilidades reais criadas, foi percebido que nem sempre o time da casa era o favorito para vencer a partida. Diante disto, foram criados três novos atributos que apontam, especificamente, quem (*home*, *away* ou *draw*) ocupa o papel do *favorite*, do *medium* e do *underdog*, como mostra a Figura 5.

	Prob. real T.C.	Prob. real E	Prob. real T.V.	Favorito	Médio	Azarão
10501	0.2518	0.2814	0.4664	a	d	h
22722	0.3922	0.3110	0.2963	h	d	a
10192	0.7038	0.1963	0.0985	h	d	a
30145	0.4167	0.2990	0.2866	h	d	a
28606	0.3038	0.3042	0.3924	a	h	d
34847	0.2078	0.2406	0.5515	a	d	h
34317	0.4260	0.3005	0.2742	h	d	a
27877	0.4964	0.2877	0.2105	h	d	a
22644	0.1696	0.2507	0.5790	a	d	h
7279	0.3909	0.2836	0.3249	h	a	d

Figura 5: Novos atributos (probabilidade real de vitória do time da casa – *home*; probabilidade real do empate – *draw*; probabilidade real de vitória do time visitante – *away*; *favorite, medium, underdog*). Elaboração própria.

A Figura 5 exibe os atributos criados que correspondem às probabilidades reais em valores decimais que são as colunas: probabilidade real do time da casa vencer a partida, probabilidade real de o empate ser o resultado final da partida e probabilidade real do time visitante vencer a partida. As colunas restantes indicam qual das opções é a: favorita da partida, a média e a menos provável, que pode-se chamar de "azarão".

Uma vez determinado o favorito, o médio e o azarão da partida, é possível derivar mais três atributos, que dizem a respeito ao acerto (*score*) individual das casas de apostas para cada uma das seleções do mercado, chamados de *hits* (*hit* do time da casa; *hit* do empate; *hit* do time visitante).

Estes atributos podem receber dois valores, 0 ou 1, a depender do acerto ou erro do resultado pela casa de apostas. Quando as casas de apostas acertam o resultado, o hit assume o valor 1 e 0, caso contrário. Existem duas formas para o hit assumir o valor 1, que são:

- Quando uma determinada seleção (time da casa, empate ou time de fora) é a favorita da partida, e o resultado final da partida for igual à seleção favorita, o hit recebe o valor 1;
- Quando uma determinada seleção (time da casa, empate ou time de fora) não for a favorita da partida, e o resultado for diferente desta seleção, o hit assume o valor 1.

Quando não acontece nenhuma das situações descritas anteriormente, o hit assume o valor 0, que são as seguintes situações:

- Se uma determinada seleção (time da casa, empate ou time de fora) for a favorita da partida e o resultado final for diferente da favorita;
- Se um determinada seleção (time da casa, empate ou time de fora) não for a favorita da partida, e o resultado final da partida for igual à seleção adotada.

Com os *hits*, é possível descobrir o escore das casas de apostas para os resultados individualmente. Para tal, basta somar todos os acertos e dividir pela quantidade de partidas. A Figura 6, mostra um gráfico gerado com a biblioteca de criação de gráficos *Plotly* [13], exibe os acertos individuais das casas de apostas. Percebe-se que as casas de apostas possuem um alto nível de acertos, principalmente para o empate e time visitante, com acertos superiores a 70%. Quanto ao time da casa, por mais que tenham vantagem sobre os adversários segundo POLLARD [14], nota-se que existe um cenário de incerteza para as casas de apostas. Por mais que elas acertem mais da metade dos casos, em 47,1% elas erram, sendo uma quantidade grande de jogos, aproximadamente 17.175 partidas.

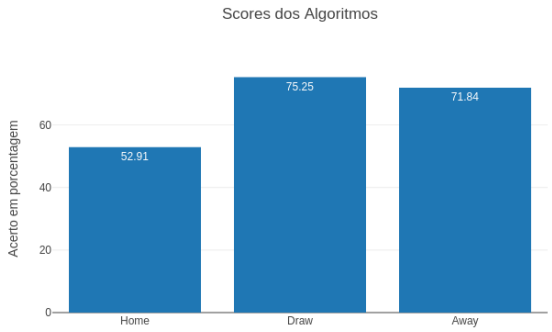

Figura 6: Escores das casas de apostas para seleções individuais do mercado *Money Line*. Elaboração própria.

Concluída a segunda etapa do processo *KDD*, antes de usar os algoritmos de aprendizagem de máquina, foi necessário dividir os dados do *data frame* em dois conjuntos: treino e teste. Para realizar a divisão, foi decidido tomar como referência os anos das competições. Assim, os dados dos anos 2006 até 2015 foram separados para treino, e os dados dos

anos 2016 e 2017 para teste, conhecido como método *Holdout* [1], que consiste em dividir os dados em dois grupos, geralmente serpa ⅔ dos dados para treino e ⅓ para testes. Após a separação dos dados do *data frame*, com o uso da biblioteca *Sklearn* [17], foram selecionados três algoritmos de aprendizagem de máquina e um modelo de regressão linear, por serem algoritmos de uso geral, que adapta-se facilmente aos dados da pesquisa, além de ser conhecidos como bons preditores. Iniciando-se a terceira etapa do processo de *KDD*: criação de um modelo para predição de resultados usando algoritmos para descoberta de padrões (mineração de dados e aprendizagem de máquina).

Algoritmos de aprendizagem de máquina

- *Random Forest*
- *Gradient Boosting*
- *AdaBoost*

Modelo de regressão linear

- *Logistic Regression*

Com os algoritmos selecionados, se faz necessário escolher os atributos para o processo de aprendizagem, sendo selecionados seis: Probabilidade real do time da casa, Probabilidade real do empate, Probabilidade real do time de fora, Seleção favorita para o resultado final da partida, Seleção mediana para o resultado final da partida e Seleção menos provável para o resultado final da partida. Além dos atributos para aprendizagem, outro foi selecionado, que é o resultado final da partida, conhecido como variável alvo. Todos estes, juntamente com os processos de aprendizagem, foram repetidos para três resultados, que são os individuais para o time da casa, o empate e o time visitante.

Para o processo de aprendizagem, foi usada a técnica de *cross validation* que consiste em realizar várias subdivisões dos dados de aprendizagem, aprendendo com uma subdivisão menor, tentando prever a seguinte, e posteriormente aprende com a que tentou prever, para tentar melhorar seu acerto na subdivisão seguinte, repetindo esse processo várias vezes até usar todos os dados reservados para treino.

Após selecionar os algoritmos, atributos e técnicas de aprendizagem, inicia-se a quarta e última etapa do processo de *KDD*: comparar diferentes técnicas de aprendizagem de máquina para serem aplicadas no modelo de predição de resultados. Cada algoritmo individualmente possui atributos que podem ser modificados, com a intenção de adequá-los da melhor forma para o contexto que ele vai ser utilizado. Assim, usando a biblioteca *Sklearn* (SKLEARN, 2018), foi aplicada uma função chamada de *Grid Search*, que permite otimizar os algoritmos testando diversas combinações de atributos (do algoritmo), retornando a melhor combinação entre eles e o melhor resultado (escore) obtido pelo algoritmo juntamente com os melhores atributos.

4 RESULTADOS

Ao processar os algoritmos de aprendizagem, foram obtidos os resultados do Gráfico da Figura 7 em que o eixo y refere-se à taxa de acerto dos algoritmos que pode estar no intervalo de 0-100, onde 0 indica que os algoritmos erraram todas as partidas, e 100 que acertaram todas. Já o eixo x são as seleções individuais (*home*, *draw*, *away*).

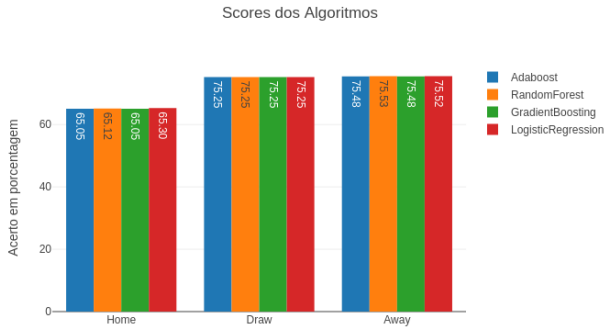

Figura 7: Resultados dos algoritmos de aprendizagem. Elaboração própria.

De acordo com o Gráfico acima, percebe-se que os algoritmos tiveram resultados semelhantes. Os resultados variaram apenas em algumas casas decimais, para as variáveis alvo (*Home* e *Away*) exceto para o empate (*Draw*). Foi decidido fazer o teste de *Wilcoxon*, que consiste em comparar duas amostras relacionadas, para testar a similaridade entre elas. Com o teste de *Wilcoxon*, deseja-se entender se os algoritmos utilizados tiveram a mesma resposta, sobre a previsão do jogo, então foram criadas duas hipóteses para serem testadas:

H0: resultados dos algoritmos são iguais
H1: resultados dos algoritmos não são iguais

Assim, foram aplicados testes entre os algoritmos, dois em dois, comparando-os para tentar provar as hipóteses. Para cada um variável alvo, seis testes foram feitos. Foi decidido fazer os testes de *Wilcoxon*, dos algoritmos para as variáveis alvo *Home* e *Away*, para reduzir a quantidade de testes a serem feitos e pelo fato do valor observado no gráfico serem exatamente iguais. Foram criadas duas tabelas contendo os *p-valores* referentes aos testes realizados.

Tabela 1: Resultados dos testes de *Wilcoxon* para o time da casa (ADA - *AdaBoost*, RF - *Random Forest*, GB - *Gradient Boosting*, LR - *Logistic Regression*).

Algoritmos	ADA	RF	GB	LR
ADA	-	1.62e-06	0.15	2.54e-10
RF	1.62e-06	-	5.73e-07	1.62e-06
GB	0.15	5.73e-07	-	7.07e-10
LR	2.54e-10	1.62e-06	7.07e-10	-

Nos testes de hipóteses, quando o *p-valor* é próximo a zero, pode-se afirmar com 95% de confiança que a hipótese *H0* não é verdadeira, onde os possíveis valores devem estar no intervalo de zero a um. Na Tabela 1, a maioria dos resultados apresentados tem o *p-valor* muito próximo a zero. Para esses casos, nega-se a hipótese *H0* e pode-se afirmar que os resultados dos algoritmos não são iguais, ou seja, por mais que o acerto alcançado pelos algoritmos sejam semelhantes, acredita-se que eles tenham acertado jogos diferentes. Porém, quando testados os algoritmos *AdaBoost* e *Gradient Boosting*, o *p-valor* assume valor alto, então com 95% de confiança, afirma-se que a hipótese *H0* não é verdadeira.

Tabela 2: Resultados dos testes de *Wilcoxon* para o time da casa (ADA - *AdaBoost*, RF - *Random Forest*, GB - *Gradient Boosting*, LR - *Logistic Regression*).

Algoritmos	ADA	RF	GB	LR
ADA	-	3.62e-40	0.99	1.39e-28
RF	3.62e-40	-	3.62e-40	9.04e-31
GB	0.99	3.62e-40	-	1.39e-28
LR	1.39e-28	9.04e-31	1.39e-28	-

Na Tabela 2, percebe-se que acontece algo semelhante ao mostrado na Tabela 1. A maioria dos resultados apresentados tem o *p-valor* próximo a zero. Para esses casos, nega-se a hipótese *H0* e pode-se afirmar que os resultados dos algoritmos não são iguais, ou seja, por mais que o acerto alcançado pelos algoritmos sejam semelhantes, acredita-se que eles tenham acertados jogos diferentes. Porém, quando testados os algoritmos *AdaBoost* e *Gradient Boosting*, o *p-valor* assume valor bem maior que zero. Assim, com 95% de confiança, afirma-se que hipótese *H0* não é verdadeira.

Como teste de *Wilcoxon*, afirma que o resultado dos algoritmos não são iguais, foi decidido usar o coeficiente de concordância *Kappa*, em que é utilizado para verificar a concordância entre dois conjuntos, independente se o conjunto é ordinal ou nominal. O valor do coeficiente de concordância de *Kappa* pode variar de 0 até 1. Quanto mais próximo de 1, maior é o indicativo de que existe uma concordância entre os conjuntos, caso contrário quanto mais próximo de 0, menor é a concordância entre os conjuntos.

Tabela 3: Resultados dos testes Kappa para o time da casa (ADA - *AdaBoost*, RF - *Random Forest*, GB - *Gradient Boosting*, LR - *Logistic Regression*).

Algoritmos	ADA	RF	GB	LR
ADA	1	0.992	0.999	0.987
RF	0.992	1	0.992	0.992
GB	0.999	0.992	1	0.987
LR	0.987	0.992	0.987	1

Observando a tabela 3, o resultado do teste *Kappa* indica que os resultados tem bastante concordância, ou seja, são bem semelhantes, detalhe que o mais uma vez os algoritmos *AdaBoost* e o *Gradient Boosting* demonstram ter o mesmo desempenho.

Tabela 4: Resultados dos testes Kappa para o time visitante (ADA - *AdaBoost*, RF - *Random Forest*, GB - *Gradient Boosting*, LR - *Logistic Regression*).

Algoritmos	ADA	RF	GB	LR
ADA	1	0.990	0.999	0.982
RF	0.990	1	0.989	0.990
GB	0.999	0.989	1	0.983
LR	0.982	0.990	0.983	1

Como esperado a tabela 4, que mostra o resultado do teste *Kappa* para os resultados do time visitante, tem a mesma interpretação que a tabela 3. Indiciando um valor bastante alto de concordância entre os resultados dos algoritmos.

5 CONCLUSÃO

Pode-se concluir que, por mais que o desempenho dos algoritmos tenham valores semelhantes, os resultados não são idênticos, segundo o teste de *Wilcoxon*, com exceção dos algoritmos *AdaBoost* e *Gradient Boosting*. Porém, após usar coeficiente de concordância *Kappa*, pode-se afirmar que os resultados dos algoritmos são bem semelhantes. Acredita-se que essa situação acontece pelo volume de dados, em que os resultados dos algoritmos são muito parecidos, mas não necessariamente são iguais.

Como o desempenho dos algoritmos tiveram resultados semelhantes, e por ter resultados parecidos com os do *AdaBoost*, o *Gradient Boosting* foi adotado para ser comparado com os resultados das casas de apostas, além de ser o algoritmo que representou melhor os resultados do time visitante.

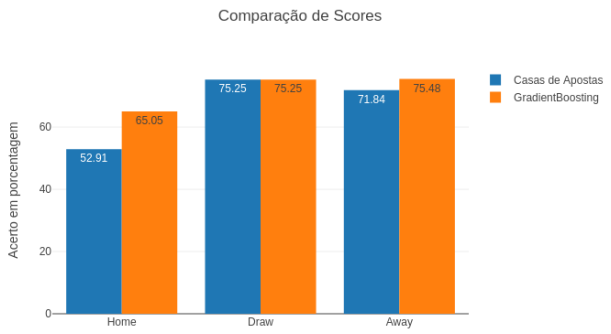

Figura 8: Comparação dos scores das casas de apostas com os scores do *Gradient Boosting*. Elaboração própria.

Pode-se observar a comparação entre o algoritmo e as casas de apostas no Gráfico da Figura 8.

Percebe-se no Gráfico que o algoritmo *Gradient Boosting* apresentou uma eficácia superior quando comparado com as previsões das casas de apostas, com destaque para o time da casa, com uma diferença de 12,14% a mais sobre as casas de apostas. Considerando que, no *data frame* usado, após se organizar e remover algumas partidas, restaram o total de 36.388 partidas, 12,14% corresponde a 4.417 partidas, ou seja, o algoritmo conseguiu acertar 4.417 resultados a mais do que as casas de apostas. As seleções do time visitante não tiveram um aumento significativo nos resultados, mas, para o time visitante, teve uma diferença de aproximadamente 3,64% totalizando 1.325 partidas. E interessante do empate é que os algoritmos e as casas de apostas tiveram o acerto com valor percentual exatamente igual, mas não pode-se afirmar que eles acertarão os mesmo jogos.

Assim conclui-se que, com os dados históricos sobre cotações das casas de apostas, ao organizá-los e processá-los, foi conseguido uma taxa de acerto maior do que a própria casa de apostas.

Trabalhos Futuros

Como os acertos dos algoritmos foram maiores que os das casas de apostas, deixa-se espaço para fazer uma análise financeira, com intenção de obter lucratividade a partir das contações oferecidas pelas casas de apostas. Pode-se ainda analisar quais foram os jogos em que os algoritmos acertaram mais que as casas de apostas, quais são as características deles e o que ele tem em comum. Por outro lado, pode-se buscar padrões nos jogos que os algoritmos erram, buscando otimizar os resultado e juntamente a possível lucratividade.

Com relação ao dados, pode-se realizar o teste de *Wilcoxon* e o *Kappa* teste entre os resultados do empate das casas de apostas e dos algoritmos, afinal eles tiveram o mesmo valor de acerto, mas não se sabe se acertaram igualmente. Adicionalmente, poderia ser construída uma nova base de dados, coletando dados de jogos de outros campeonatos, como, por exemplo: Copa do Mundo, *Champions League*, Copa Libertadores da América, dentre outros, para verificar se os algoritmos teriam bom desempenho, ou se seria mais fácil de prever. Inclusive as predições também poderiam ser feitas por campeonatos, buscando entender se existem campeonatos mais fáceis de prever, consequente mais lucrativo para apostadores, do que outros.

REFERÊNCIAS

[1] Qasem A Al-Radaideh, Emad M Al-Shawakfa, and Mustafa I Al-Najjar. 2006. Mining student data using decision trees. In *International Arab Conference on Information Technology (ACIT'2006), Yarmouk University, Jordan*. 4–5.
[2] Anaconda [n.d.]. Retrieved May 27, 2017 from https://anaconda.org/
[3] BeautifulSoup [n.d.]. Retrieved May 27, 2017 from https://www.crummy.com/software/BeautifulSoup/bs4/
[4] BetExplorer [n.d.]. Retrieved May 27, 2017 from http://www.betexplorer.com/
[5] BetFair [n.d.]. Retrieved May 27, 2017 from https://www.betfair.com/br
[6] Anthony C Constantinou. 2019. Dolores: a model that predicts football match outcomes from all over the world. *Machine Learning* 108, 1 (2019), 49–75.
[7] Usama M Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy. 1996. Advances in knowledge discovery and data mining. (1996).
[8] FIFA [n.d.]. Retrieved May 27, 2017 from http://www.fifa.com/
[9] Football-Data [n.d.]. Retrieved May 27, 2017 from http://www.football-data.co.uk
[10] Leandro Balby Marinho Igor B. Costa, Carlos Eduardo S. Pires. 2017. Sports Analytics: Mudando o Jogo. *Uberlândia - MG: Sociedade Brasileira de Computação - SBC* 1 (2017), 43–44.
[11] MySQL [n.d.]. Retrieved May 27, 2017 from https://www.mysql.com/
[12] Pandas [n.d.]. Retrieved May 27, 2017 from https://pandas.pydata.org/
[13] Plotly [n.d.]. Retrieved May 27, 2017 from https://plot.ly/python/
[14] Richard Pollard. 2008. Home advantage in football: A current review of an unsolved puzzle. *The open sports sciences journal* 1, 1 (2008).
[15] Python [n.d.]. Retrieved May 27, 2017 from https://www.python.org/
[16] Requests [n.d.]. Retrieved May 27, 2017 from http://docs.python-requests.org/
[17] Sklearn [n.d.]. Retrieved May 27, 2017 from http://scikit-learn.org/stable/
[18] SQLAlchemy [n.d.]. Retrieved May 27, 2017 from https://www.sqlalchemy.org/