Análise de dados futebolísticos para previsão de jogos

Antonio Henrique Soares,
Filipe Carlos Xavier de Oliveira,
José Augusto de Paiva Neto,
Ulysses Duarte Neves de Barros

[1]Departamento de Informática e Estatística – Universidade
Federal Rural de Pernambuco (UFRPE)
Recife – PE – Brazil

soaresantonio28@gmail.com, filcarx@gmail.com

augustopaiva4@gmail.com, ulyssesduarte14@gmail.com

Abstract. *Through the use of Data Science we will analyze football data in order to predict the outcome of a match between two Serie A teams of the Brazilian Championship, aiming to obtain profit through bets made using the prediction proposed by the algorithms. The algorithms and methods used will be described throughout this article, which will analyze which algorithm best predicts future game results. Among the methods used the use of webcrawler was essential to obtain the data that were used for machine learning including training. and validation.*

Resumo. *Através do uso de Data Science iremos analisar dados futebolísticos no intuito de prever o resultado de uma partida entre dois times da série A do Campeonato Brasileiro, visando a obtenção de lucro através de apostas feitas utilizando a predição proposta pelos algoritmos. Os algoritmos e métodos utilizados serão descritos ao longo deste artigo, onde será feita análise de qual algoritmo melhor prediz resultados futuros dos jogos, dentre os métodos utilizados o uso de webcrawler foi essencial para obtenção dos dados que foram usados para o aprendizado de máquina incluindo treinamento e validação.*

1. Introdução

Com a nova regulamentação das apostas, no Brasil, algumas empresas deste ramo começam a investir nos clubes brasileiros. Sites como Bodog, patrocinador principal da Copa do Brasil, começam a aparecer no cenário nacional.

Ainda no âmbito de apostas, o fantasy game Cartola FC é uma febre entre os torcedores brasileiros, com um número de 424 mil usuários pagantes no ano de 2018, rendendo um total de 18 milhões para o grupo Globo, mostra o interesse do público em analisar jogos e prever placares.

Muitos se arricam no mundo das apostas, usando apenas seus conhecimentos futebolisticos.Segundo um estudo da Fundação Getulio Vargas, R$4 bilhões são movimentado por ano, entre os mais de 500 sites de aposta.

Ja se dizia que o futebol é uma caixinha de surpresa, isso por que o esporte é muito imprevisivel, e uma simples característica na escalação ou como o time está se comportanto em partidas anteriores pode ser crucial para a vitória ou derrota na temporada.

Entrando na parte de utilização de dados futebolísticos temos o ótimo exemplo do campeão da Liga dos Campeões de 2018-2019, Liverpool, um dos fatores para o clube inglês conseguir este título foi como o analista do time soube se utilizar da informação advinda dos jogos, por exemplo na contratação de jogadores estrelas no time como Salah e Roberto Firmino para o time.

Para este tipo de ambiente, o entendimento da probabilidade e estatística é fundamental para sair da área de apenas um jogo de azar para algo mais real. O objetivo deste trabalho é avaliar qual algoritmo tem um melhor desempenho na predição do resultado de uma partida de futebol.

2. Trabalhos correlatos

Com a popularização do uso de tecnologia na análise de dados futebolísticos e do volume de investimento pelas partes envolvidas, desde apostadores até mesmo a comissão técnica dos clubes, vários trabalhos foram desenvolvidos utilizando diversas técnicas com o objetivo de prever o resultado das partidas ou analisar o desempenho de times e jogadores.

É comum o uso de algoritmos de aprendizado de máquina na previsão do resultado de jogos.[Nabinger 2018] apresenta uma análise das variáveis mais relevantes para um resultado positivo de uma equipe na Copa do Mundo de 2018 utilizando Árvore de Decisão, LASSO e MARS. [Schneider 2018] compara o desempenho de diversos classificadores como Random Forest, SVM e KNN, na predição do resultado das partidas da Premier League.

Existem também estudos que utilizam outras técnicas de inteligência artificial como [Cui et al. 2018], que nos apresentam uma aplicação de técnicas de programação genética para prever resultados das partidas da Premier League. Já em relação à fonte dos dados, [Gomes et al.] se difere por prever o resultado das partidas utilizando os dados de jogadores no jogo eletrônico FIFA.

3. Referencial Teórico

3.1. Mineração de Dados

Muito utilizada na área científica e no meio organizacional, a mineração de dados é algo essencial quando lidamos com um grande volume de informações ou dados não estruturados, ela nos ajuda a obter informações e fazer análises mais facilmente.

Utilizando este processo, é possível identificarmos padrões, fazer correlação entre dados aparentemente independentes, bem como tomar decisões baseadas nas informações obtidas, um caso clássico que podemos exemplificar foi a decisão de colocar dois produtos aparentemente sem relação alguma um com o outro, a cerveja e a fralda numa mesma prateleira por um dos maiores varejistas dos Estados Unidos, o resultado foi um aumento significante no número de vendas de cerveja, proporcionando um retorno de 400% sobre o investimento feito.

Em nossa análise, utilizamos a mineração de dados para a obtenção inteligente de informações, bem como analisar hipóteses inferidas de correlação sobre os dados obtidos.

3.2. Pandas

O Pandas é uma biblioteca Python para manipulação e análise de dados, utilizando estruturas e operações que simulam um banco de dados. Através do Pandas é possivél importar

arquivos de vários tipos em formato de dataframe, que consiste em uma estrutura semelhante a uma tabela, permitindo operações de select, group by, join entre outras.

3.3. Aprendizado de Máquina

A Aprendizagem de Máquina é um processo no qual os computadores desenvolvem o reconhecimento de padrões, e a partir disso podem realizar previsões sem a intervenção humana ou determinar grupos de dados com base num conjunto de exemplos de treinamento, este processo se separa em dois tipos, sendo eles:

Aprendizagem Não Supervisionada: tem como atividade a descrição onde não é dado nenhum tipo de resposta para o treinamento do algoritmo, apenas são fornecidos os exemplos que são compostos por vetores de valores sendo normalmente citados pelo nome de atributos, porém sem a resposta ou rótulo da classe associada, ficando de forma livre para identificar novos padrões nos dados e grupos a que esses dados pertencem.

Aprendizagem Supervisionada: tem como atividade a predição onde são dados vetores de atributos e classes predefinidos para o treinamento do algoritmo, com o objetivo de aprender a regra acerca do problema. Dentro deste tipo de aprendizagem ainda há a subdivisão em algoritmos que a partir dos atributos rotulam classes discretas sendo declarados como de classificação e para valores contínuos como regressão. [Monard and Baranauskas 2003]

A partir desta descrição são necessários a explicação de alguns conceitos para complementar o que é a aprendizagem supervisionada, são eles:

Exemplo: podendo ser denominado de caso, instância ou padrão é um vetor de atributos ou valores que descreve o objeto a ser analisado.

Atributo: descreve uma característica de um exemplo, tendo comumente dois tipos que são o nominal, onde não há preocupação com ordem nos valores(um exemplo seria a raça de um cão: pug, poodle, labrador) e o contínuo onde existe uma ordem nos valores(exemplo: salário de um trabalhador).

Classe: é o atributo que rotula o objeto de interesse ou exemplo, sendo onde se procura poder efetuar predições e como um atributo pode ser nominal ou real.

Acurácia: métrica utilizada quando o classificador está treinado pegando como teste de desempenho um conjunto de exemplos previamente separados sendo um indicativo da taxa de acerto ou precisão do algoritmo.

Overffiting e Underfitting: Quando a hipótese sobre o exemplo induzida se ajusta muito pouco ao conjunto de exemplos de treinamento se diz que houve um Underfitting, e quando a hipótese se ajusta em excesso ocorre o Overfitting, ter overfitting implica em o seu classificador não conseguir se adaptar a dados novos, dando a ideia que o seu algoritmo decorou as respostas e não aprendeu como o desejado.

Referente a toda essa explicação sobre o que é um aprendizado de máquina temos um exemplo fácil de uso dela que é em relação à análise de concessão de crédito de um banco, dependendo das informações dadas pela pessoa ao solicitar um cartão de crédito, a aprendizagem permite com que se tenha uma análise baseado na sua base de dados se o cliente será um bom pagador ou não e com isso decidir se o cartão será liberado. Neste artigo será utilizado a aprendizagem de máquina de modo supervisionado e dos tipos de

classificação e regressão.

3.3.1. Random Forest

Trata-se de um algoritmo de aprendizagem supervisionada e a maneira como o algoritmo funciona é basicamente utilizando a combinação de várias árvores de decisão, que dada algumas configurações (features, hiperparâmetros...) tornam seus resultados mais assertivos. Sua versatilidade o torna um algoritmo muito bom para ser utilizado por pessoas que desejam iniciar um processo de aprendizado tanto com problemas de classificação quanto de regressão. Apesar de utilizar árvores de decisão, a maneira como são feitos os seus processos é diferente, como o próprio nome sugere, alguns processos são feitos de forma aleatória, essa característica quando se soma com outros fatores, levam o algoritmo a evitar overfiting, como pode ocorrer no caso de árvores de decisão muito profundas. [Breiman 2001]

Uma etapa fundamental do processo é trabalhar na normalização dos dados que serão utilizados. O processo de normalização é importante pois algo comum no aprendizado de máquina é que quanto mais características a cerca de um problema o algoritmo carrega, maior a probabilidade de ocorrer overfitting. Utilizando a biblioteca SKLearn, que contém vários algoritmos e funcionalidades relacionadas ao aprendizado de máquina, o processo de normalização é feito facilmente, logo após o treinamento do algoritmo a biblioteca permite que possamos identificar quais dados tem maior importância, com isto podemos retirar os menos importantes ou aqueles que não julgamos tão relevantes para o nosso problema.

Após feita a normalização, é interessante que façamos pequenas mudanças nos hiperparâmetros a fim de identificar uma boa configuração que retorne bons resultados a cerca do nosso problema, alguns deles são: n_stimators, max_features, min_sample_leaf, n_jobs, random_state, oob_score. Para um maior entendimento a cerca desses parâmetros é importante que se faça uma leitura da documentação.

3.3.2. Support Vector Machine

O SVM como também é chamado, é mais um algoritmo que compõe o grupo dos algoritmos supervisionados. Muito utilizado em problemas de classificação, mas também pode ser utilizado em problemas de regressão, funciona bem em ambos.

Neste algoritmo, o objetivo é que encontremos um hiper-plano que faça a melhor distinção possível das classes utilizadas, em outras palavras, cada informação é plotada num gráfico e o algoritmo busca aquilo que está entre elas, buscando separar as classes de dados.

Assim como todo algoritmo possui suas vantagens e desvantagens. Este algoritmo trabalha melhor quando temos dados bem definidos e separados, mas não funciona muito bem quando temos um grande volume de dados, pois seu tempo de execução aumenta consideravelmente de uma forma não proporcional.

3.3.3. Gradient Boosting

Esta técnica surgiu através do aprimoramento da técnica AdaBoost que inicialmente tratava de problemas padrões de aprendizado de máquina, ao longo do tempo AdaBoost vem sendo aprimorada e outras vertentes da técnica surgiram, incluindo o Gradient Boosting que busca promover um boosting (aprimoramento) dos algoritmos de aprendizado de máquina. [Mayrink 2016]

O algoritmo tem como objetivo a redução do erro de previsão, para isto, ele faz diversas iterações sobre uma previsão inicial baseada na média dos resultados obtidos com a amostra de dados utilizadas para treinamento, as iterações cessam quando uma condição é satisfeita ou um número máximo é atingido.

Assim como em outros algoritmos de aprendizado há a possibilidade de overfitting e ela deve ser evitada, no caso do Gradient, o overfitting pode ocorer principalmente no caso em que o algoritmo é executado por um tempo prolongado, portanto deve-se utilizar alguma estratégia para evitar que isto ocorra, uma estratégia bastante utilizada é de reservar parte dos dados de treinamento para validação do algoritmo, caso alguma exceção ocorra durante esta fase de validação, todo o processo deve ser interrompido.

3.3.4. Multilayer Perceptron

O Multilayer Perceptron trata-se de uma rede neural que normalmente é composta por: uma entrada de dados, uma ou mais camadas e uma ou mais saídas, tudo depende do problema, tudo irá depender do problema que se propõe a resolver.

É comum que a entrada de dados seja numérica quando tratamos de classificação, mas nada impede o Multilayer Perceptron de receber multi classes, contanto que haja a categorização dos dados, ou seja, cada elemento não numérico será convertido em números que o algoritmo possa reconhecer.

As camadas são etapas pela qual os dados passam sofrendo ajustes, ou reajustes quando se utiliza o backpropagation, uma técnica de ida e volta dos dados.

A saída ocorre quando as condições do algoritmo são satisfeitas ou quando ocorre um número máximo de iterações, no caso de haver mais de uma saída é necessário a avaliação do que melhor aborda o seu problema, algo comum é a utilização da média das saídas do algoritmo.

4. Metodologia

Primeiro desafio deste trabalho será buscar uma base de dados grande o suficiente para conseguir ser executada nos algoritmos sem perigo de overfitting e suficientemente grande para conseguir extrair dados para melhor entendimento do campeonato.

Para isto, foi feito um crawler no site fbref.com, que tem possui diversos campeonatos, mas neste trabalho foi focado apenas no campeonato brasileiro, entre os anos de 2015 e 2019.

Para as análises de dados foi utilizada a distribuição anaconda, que possui diversas ferramentas como bibliotecas para aprendizado de máquina e tratamento de dados como:

pandas, numpy e sklearn, e uma IDE (Spider) que auxilia na visualização de datasets.

5. Estrutura dos Dados

Os dados foram armazenados em arquivos de formato csv onde cada linha do arquivo são as características do time em uma determinada partida.

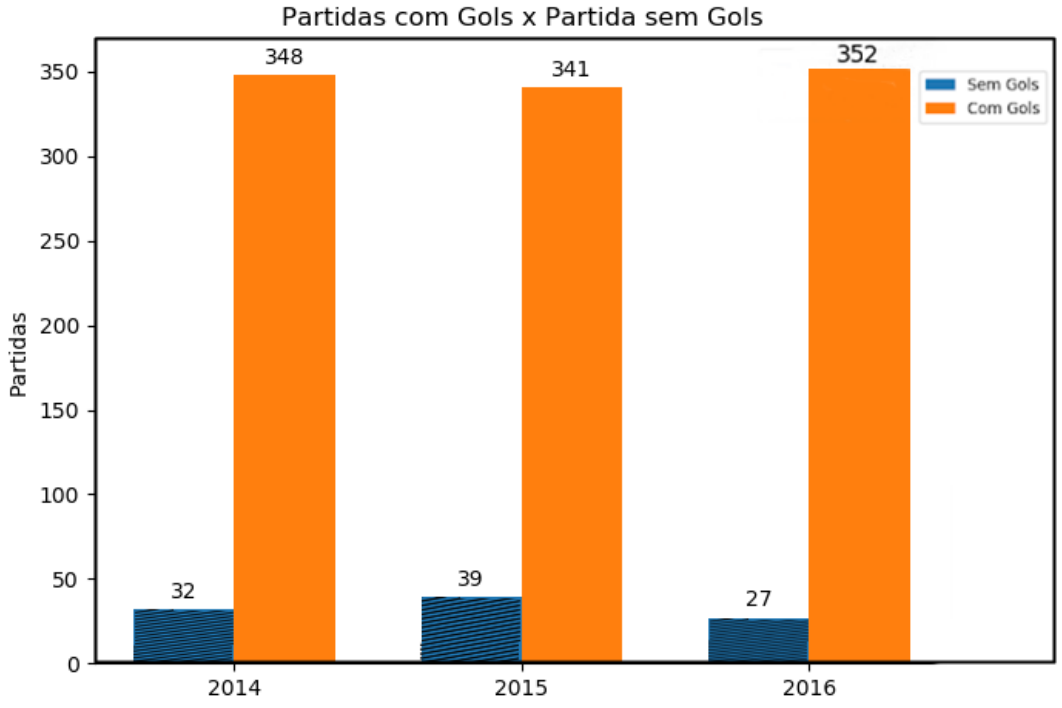
Os dados consistiam em Rodada, Código identificador do Time, Nome do Time, Cartões Amarelos, Cartões Vermelhos. Posse de Bola, Toques na Bola, Total de Passes, Passes Certos, Total de Chutes, Chutes no Alvo, Lançamentos Laterais, Roubadas de Bola, Impedimentos, Batalhas Aéreas Vencidas, Defesas do Goleiro, Interceptações, Faltas, Cruzamentos, Escanteios, Lançamentos, Tiro de Meta, Público do Estádio, Chutões, Gols.

Na leitura destes dados apenas se obtiam os de tipo numérico, sendo agrupados por partida em um dataframe pandas, fazendo com que tanto as informações do time mandante quanto a informação do time visitante estivessem na mesma linha. Neste dataframe foi necessária a criação de uma atributo vencedor indicando se foi o mandante que venceu, se o visitante que venceu ou se deu empate a partida, é este o atributo que dá um rótulo de classe a partida.

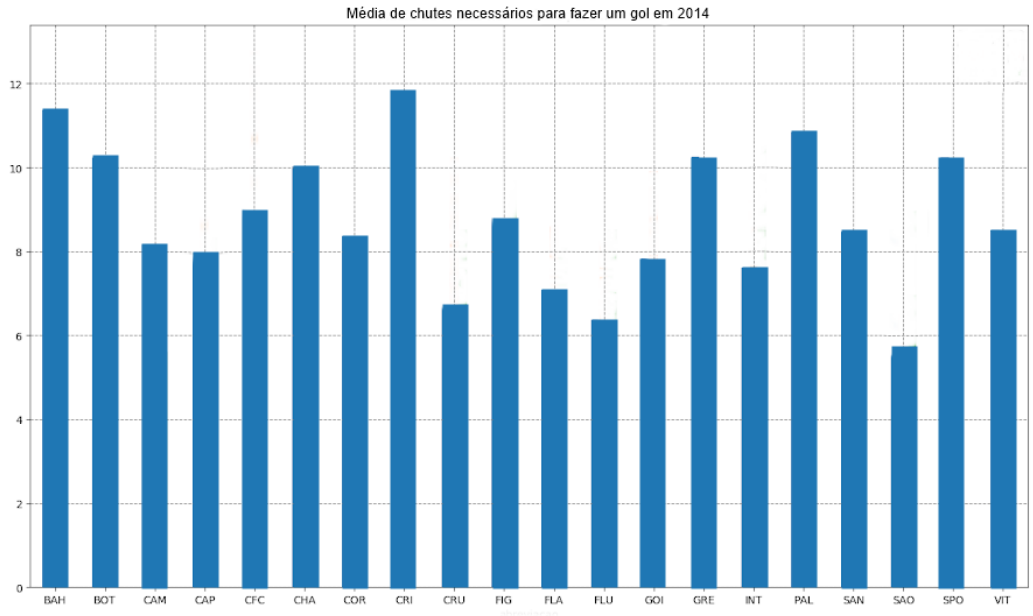
6. Análise dos dados

Através do crawler no site fbref.com, foi possível obter dados de cada partida do campeonato brasileiro entre os anos de 2015 e 2019, com isso foi possível realizar uma análise inicial e consequentemente obter algumas informações.

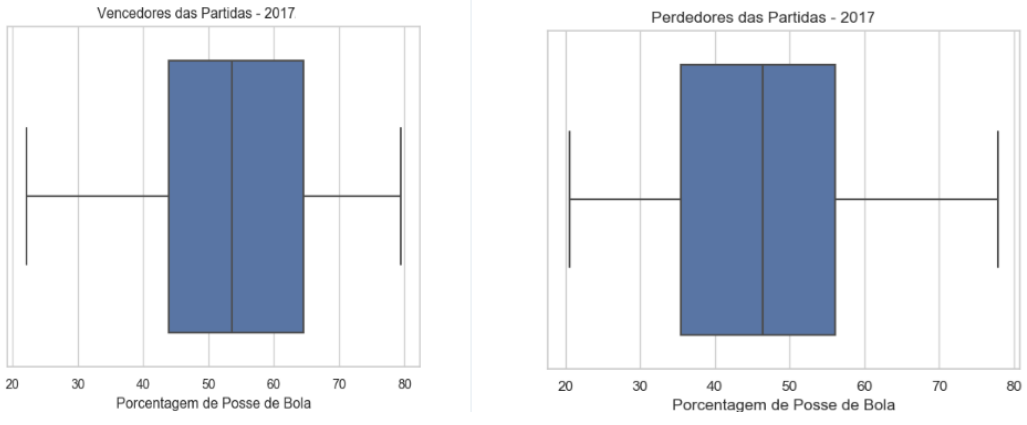
Inicialmente foi observada uma diferença entre o número de partidas que tiveram gols e o número de partidas que não tiveram gols, como exibido no gráfico abaixo, entre 2014 e 2016 a quantidade de jogos sem gols correspondeu entre 7% e 10%.



Além disso, foi analisada a quantidade de chutes eram necessários para que um time marcasse um gol. No gráfico abaixo é possível notar que os times que terminaram o ano de 2014 nas últimas colocações, como Criciúma e Bahia, precisavam tentar mais vezes para acertar o gol enquanto times no topo da tabela precisavam tentar bem menos, como foi o caso do Cruzeiro.



A partir desses dados também foi possível expor a diferença entre a posse de bola entre os times que venceram as partidas e os que perderam as partidas no campeonato brasileiro de 2017, como exposto no gráfico abaixo.

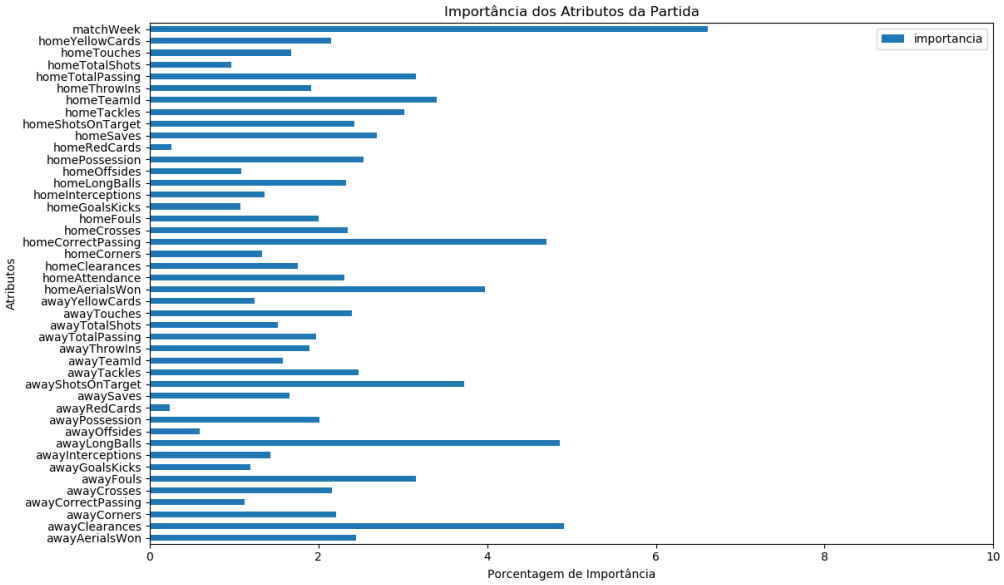


7. Importância dos dados

Para obter essa informação foi utilizado o algoritmo Random Forest do tipo de classificação com os arquivos dos anos de 2016 a 2019 como entrada, onde dentro de cada arquivo continham as partidas respectivas do ano ao qual o arquivo se refere.

Como próximo passo foi executado o treinamento do algoritmo, baseado nesse treinamento é que foi possível gerar o dado de importância dos atributos, por que a partir

deste treino o algoritmo percebe qual atributo pesa mais em sua decisão do resultado da partida.

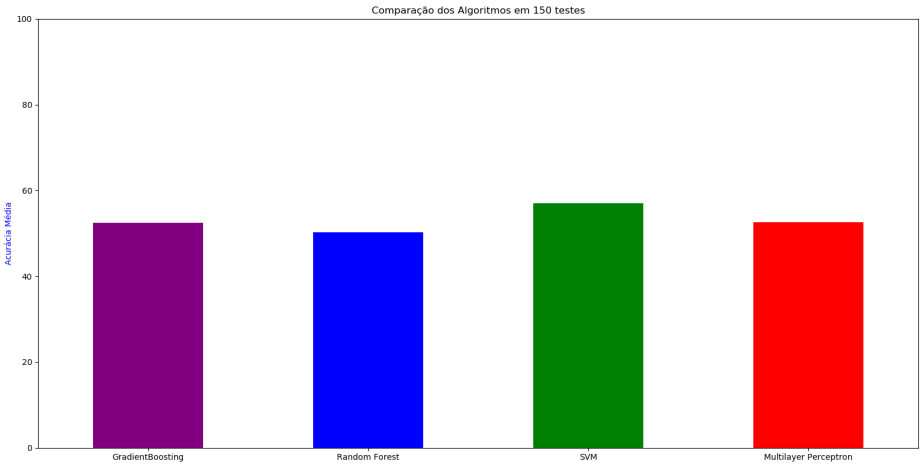


8. Resultados

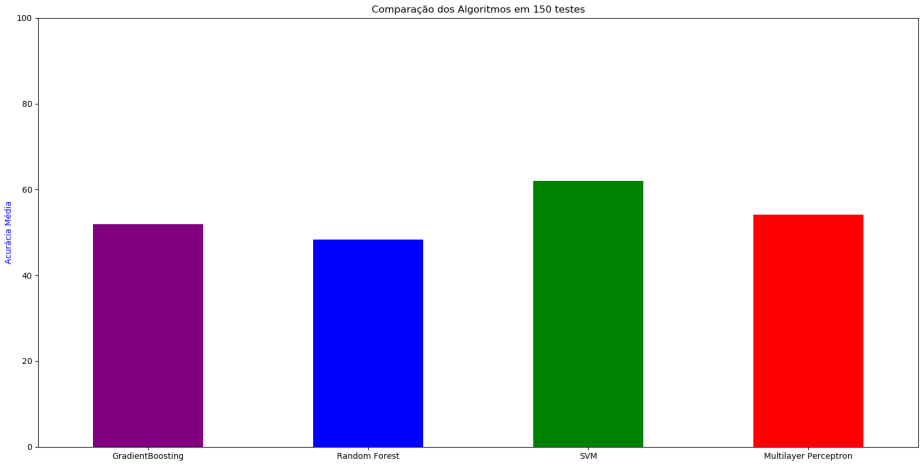
A partir da leitura dos arquivos CSV dos anos 2016 a 2019, foram obtidas 1074 partidas, sendo a sua distribuição por classe de 392 partidas com vitória do mandante, 341 partidas com vitória do visitante e 341 partidas que houveram empate. De forma a evitar um viés dos algoritmos para a classe com maior quantidade, foi colocado como fixo a quantidade de 341 que corresponde a classe com menor quantidade de partidas, e feita a divisão de 80% desses dados para treino e 20% para teste.

Com base no gráfico da taxa de importância dos stats da partida iremos rodar os algoritmos de Random Forest, SVM, GradientBoosting e Multilayer Perceptron, com variações nos stats de partida para identificar qual conjunto de stats irá promover a melhor acurácia de classificação.

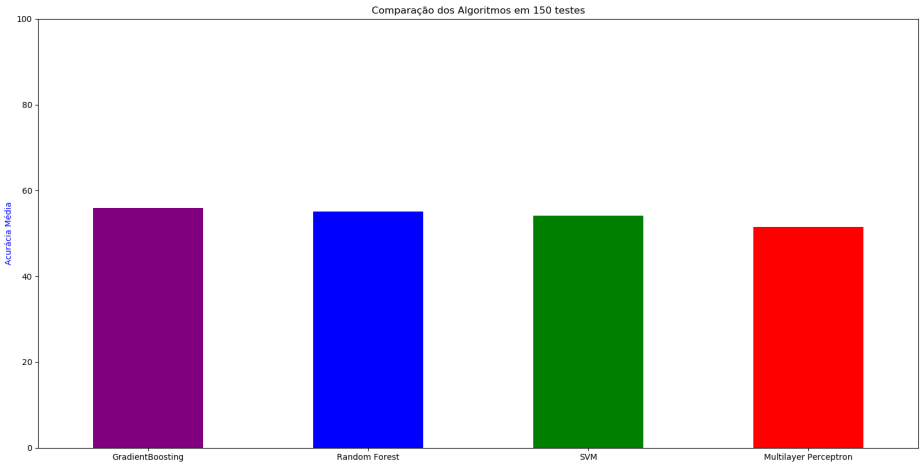
Usando todos os stats que nossa base de dados oferece, percebemos que o algoritmo SVM se sobressai com uma acurácia média de 57,07% em seguida fica o Multilayer Perceptron com 52,63 %, GradientBoosting com 52,46% e por último o Random Forest com 50,24%. Segue o gráfico abaixo para melhor visualização.



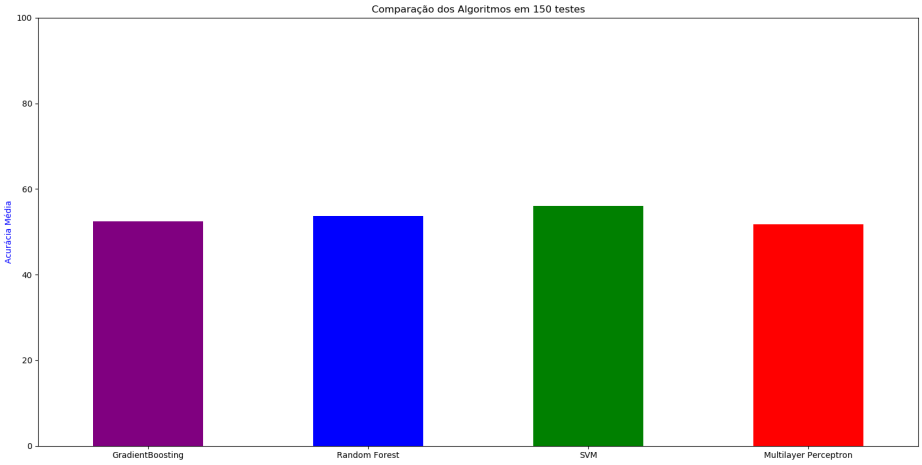
Fazendo uso de apenas os stats de partida com relevância maior do que 1%, retirando os stats de cartão vermelho, houve uma significativa melhora de acurácia, com o SVM ainda como o melhor dos algoritmos com 61,95%, em seguida dos outros na mesma ordem do teste anterior



Neste caso estamos usando apenas stats com nível de relevância superior a 1,5%, neste caso o que melhor saiu foi o GradientBoosting mas com uma acurácia média pior do que o SVM no teste anterior.



No quarto teste foi usado apenas stats com relevância superior a 2%, neste caso o SVM volta ser o melhor algoritmo mas com média ainda melhor do que o teste 2.



9. Trabalhos futuros

O que foi feito até aqui é uma classificação de vencedor baseado nos status da partida a ser classificada. Um dos objetivos do projeto é conseguir predizer o vencedor de uma partida que ainda vai acontecer, sem ter dados da partida, apenas as informações das partidas anteriores da temporada.

Uma outra linha de pesquisa é avaliar os jogadores em suas partidas, para se ter uma análise de desempenho dos jogadores ao longo do tempo a fim de conseguir predizer vencedor de uma partida baseado na sua escalação, o que seria uma previsão mais realista.

Muitos trabalhos de previsões de jogos já foram feitos, cada um usando uma linha de pesquisa e uma ideia do que seria ideal para se classificar. Um dos planos de pesquisa a ser feito é criar um algoritmo que faça uso de todas essas linhas a fim de criar um modelo de previsão que use a qualidades de cada um e anule a ineficácia com a qualidade de

outro trabalho. Estes trabalhos serão continuados em projeto de finalização de cursos dos estudantes deste artigo.

10. Conclusão

A análise de dados no mundo do esporte está sendo cada vez mais usado para contratação, análise dos jogadores internos do time e entender em qual fase o jogador está para contratação ou venda.

Uma área que também está crescendo no Brasil são as apostas de jogo, então entender e analisar os dados do futebol a fim de entender a possibilidade de vitória em um confronto é de muita utilidade para quem quer trabalhar com dados.

Com base nos algoritmos utilizados para este trabalho, o SVM se apresentou como a melhor solução para classificar o vencedor de partida com base nos stats do jogo, e com uma taxa de relevância superior a 1%, taxa de relevância mostrada no tópico 7.

Referências

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Cui, T., Li, J., Woodward, J. R., and Parkes, A. J. (2018). An ensemble based genetic programming system to predict english football premier league games. *In Evolving and Adaptive Intelligent Systems (EAIS), 2013 IEEE Conference on, pages 138–143. IEEE*.

Gomes, B. G., Moreira, M. C. G., and Holanda, P. H. F. Treinamento supervisionado para previsao de partidas de futebol: Uma abordagem usando dados de videogames. 1(1):6.

Mayrink, V. T. d. M. (2016). Avaliação do algoritmo gradient boosting em aplicações de previsão de carga elétrica a curto prazo.

Monard, M. C. and Baranauskas, J. A. (2003). Conceitos sobre aprendizado de máquina. *Sistemas inteligentes-Fundamentos e aplicações*, 1(1):32.

Nabinger, A. M. (2018). Utilizacao de algoritmos do tipo machine learning supervisionado para a caracterizacao dos resultados da copa do mundo de futebol de 2018. 1(1):44.

Schneider, C. F. (2018). Machine learning aplicado na previsão de resultados de partidas de futebol : um estudo de caso para comparação de diferentes classificadores. 1(1):93.