**Técnicas de Machine Learning Aplicadas à Previsão de Jogos de Futebol Através de Bibliotecas Python**

**Thiago Júnio Saraiva¹, Diego Roberto Gonçalves de Pontes²**

1Aluno de Graduação do Curso de Ciência da Computação da PUC Minas *Campus* de Poços de Caldas – MG, Brasil

2Orientador no Curso Ciência da Computação da PUC Minas *Campus* de Poços de Caldas – MG, Brasil

thg\_1988[@hotmail.com, diego.pontes@pucpcaldas.br](mailto:renanferreirameira@hotmail.com)

***Abstract.*** *This article examines the use of Python libraries in applying Machine Learning to predict football games based on past information. Through the use of prediction algorithms such as Random Forest, SVC e Logic Regression combined with exploratory analysis techniques, where comparisons between methods are presented in order to demonstrate their effectiveness and limitations. The results obtained show that it is possible to obtain a relatively high accuracy rate even with a certain limitation of the data used, concluding that the chance of success is closely related to the robustness of the information used.*

***Resumo.*** *Este artigo aborda o uso de bibliotecas Python na aplicação de Machine Learning para previsão de jogos de futebol baseando-se em informações passadas. Através do uso de algoritmos de previsão como Random Forest, SVC e Logic Regression aliado às técnicas de análise exploratória, são apresentadas comparações entre os métodos a fim de demonstrar suas eficácias e limitações. Os resultados obtidos mostram que é possível obter uma taxa de precisão relativamente alta mesmo com certa limitação dos dados utilizados, concluindo-se que a chance de acerto está intimamente relacionada à robustez das informações usadas.*

# **INTRODUÇÃO**

A previsão de resultados em jogos de futebol é uma área de grande interesse tanto para entusiastas do esporte quanto para aqueles que buscam melhorar suas chances em sites de apostas. Com o avanço das técnicas de aprendizado de máquina, tornou-se possível utilizar algoritmos sofisticados para analisar grandes volumes de dados e mensurar previsões mais precisas. Este artigo tem como objetivo explorar de forma detalhada o funcionamento de algumas das principais bibliotecas da linguagem Python voltadas para esse tipo de evento, utilizando as técnicas a fim de atingir uma boa precisão nas previsões de partidas de futebol.

Ao longo deste artigo, será abordado o conceito de cada um desses algoritmos e como eles são implementados nessa abordagem específica. Além disso, será apresentado exemplos práticos de como esses modelos podem ser treinados e testados com dados reais de partidas de futebol, com o intuito de fornecer *insights* valiosos acerca das informações obtidas. O objetivo deste projeto é alcançar a maior taxa de acerto de resultado de jogos possível através do uso desses métodos, possibilitando a monetização com os resultados obtidos.

# **APRENDIZADO DE MÁQUINA**

O Aprendizado de Máquina (*Machine Learning*, em inglês) é um subcampo da Inteligência Artificial (IA) que se concentra no desenvolvimento de algoritmos e técnicas que permitem aos computadores aprender e fazer previsões ou decisões baseadas em dados. Em vez de ser explicitamente programado para executar uma tarefa, um sistema de aprendizado de máquina usa padrões e inferências a partir de dados para melhorar seu desempenho em tarefas específicas, simulando o comportamento humano.

# 

# **2.1 VISÃO GERAL DO APRENDIZADO DE MÁQUINA**

O Aprendizado de Máquina clássico depende mais da intervenção humana para aprender, conhecido como aprendizado supervisionado, onde são determinados os conjuntos de recursos para entender as diferenças entre as entradas de dados, geralmente exigindo dados mais estruturados para aprender. Com a evolução das técnicas de *Machine Learning*, surgiu um novo termo chamado de Aprendizado Profundo (*Deep Learning*). Essa técnica possibilita o uso de dados não estruturados, em sua forma bruta (por exemplo, texto ou imagens), podendo determinar automaticamente o conjunto de recursos que distinguem diferentes categorias de dados umas das outras. É um artifício que utiliza as Redes Neurais Artificiais (ANNs), que basicamente são algoritmos que simulam o funcionamento de um neurônio humano.

Os algoritmos de aprendizado de máquina podem ser classificados em três categorias:

1. Aprendizagem supervisionada: É definido pelo uso de conjuntos de dados rotulados para treinar algoritmos a fim de classificar dados, como por exemplo definir se um e-mail é ou não *spam* com base em algumas informações. Alguns métodos usados no aprendizado supervisionado incluem redes neurais, *Naive Bayes,* *Linear Regression*, *Logistic Regression*, *Random Forest* e *Support Vector Machine* (SVM);
2. Aprendizagem não supervisionada: Usam algoritmos de aprendizado de máquina para analisar e agrupar conjuntos de dados não rotulados. Esses algoritmos descobrem padrões ocultos ou agrupamentos de dados sem a necessidade de intervenção humana. Algoritmos usados nessa abordagem incluem redes neurais, agrupamento k-means e métodos de agrupamento probabilísticos;
3. Aprendizagem semi supervisionado: O aprendizado semi supervisionado oferece um meio termo entre os dois anteriores. Durante o treinamento, ele usa um conjunto de dados rotulado menor para orientar a classificação e a extração de recursos de um conjunto de dados maior e não rotulado[1];

**3. ALGORITMOS DE APRENDIZADO DE MÁQUINA SUPERVISIONADO**

Conforme mencionado, existem diversos tipos de algoritmos de *Machine Learning*, cada um com suas características intrínsecas, com vantagens e desvantagens a serem consideradas de acordo com a necessidade, embora possam existir determinados problemas onde será possível utilizar vários desses algoritmos, a fim de comparar o desempenho entre eles. Abaixo estão listados os algoritmos utilizados no desenvolvimento da proposta.

# **3.1 SUPPORT VECTOR CLASSIFICATION (SVC)**

É um algoritmo de aprendizado supervisionado primordialmente utilizado em desafios de classificação. O SVC faz parte dos Support Vector Machines (SVMs), que são poderosos modelos de aprendizado de máquina usados para resolver problemas complexos de classificação e regressão. Basicamente, essa técnica consiste em analisar os dados através de vetores gerados pela disposição desses dados num plano dimensional, de forma que seja possível classificar os dados separando-os por uma linha ou hiperplano. Seus pontos fortes são a possibilidade de lidar com problemas multidimensionais, ser eficiente em termos de memória por utilizar subconjuntos de pontos de treinamento (vetores de suporte) na função de decisão e também a capacidade de classificar dados lineares e não lineares, graças às funções kernel, que são artifícios desse algoritmo. Os pontos fracos são o fato de não ser adequado para conjuntos de dados muito grandes devido à complexidade computacional, ser menos eficaz quando há muito ruído nos dados e também a necessidade de selecionar corretamente o kernel e os parâmetros utilizados, que dependerá da disposição dos dados [2].

**3.2** **RANDOM FOREST**

O *Random Forest* é um algoritmo de *Machine Learning* baseado em um outro algoritmo chamado *Decision Trees*, que são modelos de aprendizado supervisionado utilizados tanto para classificação quanto para regressão. Elas funcionam dividindo iterativamente o conjunto de dados em subconjuntos mais homogêneos de acordo com certas condições baseadas nos atributos dos dados. As *Decision Trees* funcionam como se fosse um fluxograma com várias estruturas de decisão, sendo um algoritmo muito útil para determinadas tarefas que envolvem dados com categorias de classificação múltiplas, dados nominais e até mesmo informações fora do padrão. Porém o ponto negativo fica por conta do *Overfitting*, que é o fato de se ajustar excessivamente aos dados de treinamento e também ao risco de instabilidade em caso de pequenas variações nos dados, resultando em dados completamente diferentes. Baseado nas limitações das *Decision Trees* que foi desenvolvido o algoritmo *Random Forest*, pois é justamente a aleatoriedade dos dados que vai prevenir que os resultados sofram com o *Overfitting*, já que os resultados de algumas árvores podem ser descartados, enquanto outros poderão até se repetir, fazendo com que os dados sejam mais confiáveis e precisos.[3].

**3.3** **LOGISTIC REGRESSION**

Logistic Regression é um algoritmo derivado do *Linear Regression*, um dos primeiros algoritmos e mais conhecidos que foram utilizados em *Machine Learning* devido à sua simplicidade e eficácia [4]. Ambos algoritmos são ideais para lidar com dados onde as informações possuem certa correlação entre si, utilizando-se do aprendizado supervisionado para prever determinados padrões. Funcionam de forma a compreender a relação entre uma variável dependente (ou variável alvo) e uma ou mais variáveis independentes (ou variáveis preditoras). A grande diferença entre eles está na variável dependente (resultado), onde na Regressão Linear pode gerar inúmeros resultados diferentes, enquanto a Regressão Logística possui uma lista limitada de resultados, geralmente somente duas (sim ou não, zero ou um, verdadeiro ou falso).

# **4. ESTUDO DE CASO**

Conforme apresentado anteriormente, os algoritmos selecionados para a realização deste estudo foram os métodos *Support Vector Classification (SVC)*, *Random Forest* e *Logistic Regression*. Serão apresentados tudo o que foi envolvido nos experimentos, como *Dataset*, *Features*, análise exploratória, parâmetros, resultados e comparações entre os métodos.

# 

# **4.1 DATASET**

Os dados analisados são informações referentes aos jogos do Campeonato Brasileiro desde 2012 até 2024. Logo, estão sendo utilizadas informações das doze últimas temporadas desse torneio, somando um total de 4635 jogos, 22 colunas e 36 equipes que participaram ao longo das temporadas, sendo utilizadas as *Features* Home, Away, HG, AG, Res, PH, PD, PA, MaxH, MaxD, MaxA, AvgH, AvgD, AvgA, que correspondem respectivamente às informações de time da casa, time visitante, gols do time da casa, gols do time visitante, quem venceu a partida, as probabilidades de vitória, empate e derrota, as probabilidades máximas de vitória, empate e derrota e também a médias das probabilidades de vitória, empate e derrota [5].

# **4.2 RESULTADOS**

Para os três modelos utilizados, o *Dataset* foi separado para treinamento e teste de forma manual, visto que a ordem cronológica dos dados são relevantes. Como os dados já se encontravam ordenados por data de acontecimento de forma ascendente, foi usado as informações da linha 1 até a linha 3041 para treinamento das *Features* e das *Labels*. Para os testes deste treinamento, foram realizados os testes de previsão da linha 3042 até a linha 4181. Por fim, para a previsão final dos dados após os treinamentos e testes, foi usado os registros da linha 4182 até a linha 4636. No experimento também foi utilizado a ferramenta *SelectKBest* para definir as melhores *Features* a serem usadas, para dessa forma se conseguir melhores resultados. As informações detalhadas sobre cada um dos modelos são apresentadas abaixo.

# 

# **SUPPORT VECTOR CLASSIFICATION (SVC)**

Para esse modelo foram utilizados o algoritmo tanto em sua forma padrão quanto na utilização dos *Hyper Parameters*, que se trata de uma técnica para realizar testes modificando os parâmetros e depois realizar uma comparação entre os resultados obtidos para assim definir qual o melhor parâmetro a ser usado. Abaixo seguem os resultados obtidos.

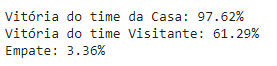
Taxa de precisão nos testes sem *Hyper Parameters*:



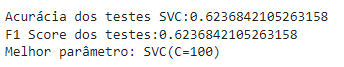
Taxa de precisão final sem *Hyper Parameters:*



Porcentagem de acertos final para cada resultado sem *Hyper Parameters*:



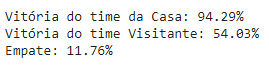
Taxa de precisão nos testes com *HyperParameters*:



Taxa de precisão final com *HyperParameters*:



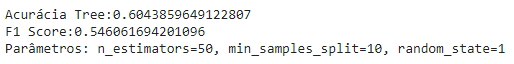
Porcentagem de acertos final para cada resultado com *HyperParameters*:



# **RANDOM FOREST**

Para o modelo foram utilizados primeiramente os parâmetros padrões e posteriormente feitos modificações nos parâmetros de forma manual, a fim de comparar qual seria a melhor combinação de valores, já que a ferramenta *HyperParameters* não se aplica a este algoritmo. Abaixo seguem os resultados obtidos tanto com a forma padrão como utilizando o melhor parâmetro encontrado nos testes de comparação.

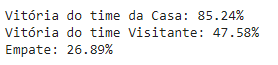
Taxa de precisão nos testes com parâmetros padrões:



Taxa de precisão final com parâmetros padrões:



Porcentagem de acertos final para cada resultado com parâmetros padrões:



Taxa de precisão nos testes com os melhores parâmetros identificados:



Taxa de precisão final com os melhores parâmetros identificados:



Porcentagem de acertos final com os melhores parâmetros identificados:



# **LOGISTIC REGRESSION**

Para esse modelo também foram utilizados o algoritmo tanto em sua forma padrão quanto na utilização dos *Hyper Parameters*, que se trata de uma técnica para realizar testes modificando os parâmetros e depois realizar uma comparação entre os resultados obtidos para assim definir qual o melhor parâmetro a ser usado. Abaixo seguem os resultados obtidos.

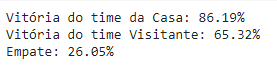
Taxa de precisão nos testes sem *Hyper Parameters*:



Taxa de precisão final sem *Hyper Parameters:*



Porcentagem de acertos final para cada resultado sem *Hyper Parameters*:



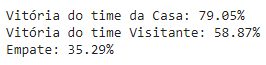
Taxa de precisão nos testes com *HyperParameters*:



Taxa de precisão final com *HyperParameters*:

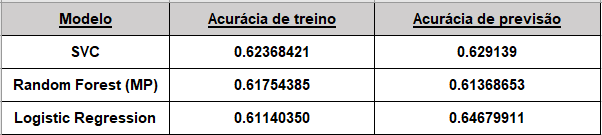


Porcentagem de acertos final para cada resultado com *HyperParameters*:



# **4.3 PORCENTAGEM DE ACERTOS FINAL**

De acordo com os resultados obtidos apresentados na Tabela 1, pode-se verificar que todos os modelos tiveram uma taxa de acerto geral muito próxima, com exceção do *Logistic Regression* que teve um melhor desempenho nas previsões finais. Contudo, alguns algoritmos se mostraram melhor para prever vitórias do time da casa (*SVC* com 97% de acerto), para prever vitória do time visitante (*Logistic Regression* com 65% de acerto) e também para prever os empates entre as equipes (*Logistic Regression* com 35% de acerto). O algoritmo *Random Forest* foi o que obteve o pior desempenho dos três.



**Tabela 1**: Resultado final com a comparação entre os modelos utilizados.

# **5. CONCLUSÃO**

Diante do que foi apresentado nos testes, pode-se concluir que os modelos de classificação e regressão apresentados podem ser utilizados como bons preditores de resultados, desde que tenham boas *Features* e seja verificado os melhores parâmetros. Para o fim específico de previsão de jogos de futebol, os modelos se mostraram excelentes para prever as vitórias do time da casa e bons para prever vitória do time visitante, não se mostrando eficientes para previsões de empates.

Foi observado também que o uso das probabilidades de *odds* das casas de apostas são boas *Features* a serem consideradas, visto que elas resumem o sentimento de presunção dos apostadores, que geralmente levam em conta todos os atributos envolvidos nas partidas, como momento de uma equipe, valor de mercado, desempenho dos jogadores, tradição, histórico de confronto, entre outras informações relevantes, acabando por resumir estes atributos através dos percentuais de vitória, derrota ou empate antes das partidas de futebol, não sendo necessário a utilização dessas informações como *Features* de um Dataset desse tipo.

# **REFERÊNCIAS**

1. IBM [homepage na Internet]: O que é aprendizado de máquina (ML). Disponível em: <https://www.ibm.com/br-pt/topics/machine-learning>, acesso em 05/2024.
2. Segatto, Vinicius [homepage na Internet]: SVM, ou Support Vector Machine. Disponível em: [https://medium.com/liga-mackenzie-de-ia-ci%C3%AAncia-de-dados/svm-ou-support-vector-machine-7efcabdcc7be, acesso em 06/2024](https://medium.com/liga-mackenzie-de-ia-ci%C3%AAncia-de-dados/svm-ou-support-vector-machine-7efcabdcc7be).
3. IBM [homepage na Internet]: O que é uma Árvore Aleatória. Disponível em: <https://www.ibm.com/br-pt/topics/random-forest>, acesso em 05/2024.
4. Damasceno, Laura [homepage na Internet]: Regressão Linear? Disponível em: <https://medium.com/@lauradamaceno/regress%C3%A3o-linear-6a7f247c3e29>, acesso em 06/2024.
5. Históricos de jogos do Campeonato Brasileiro 2012-2024. Disponível em: [https://football-data.co.uk/brazil.php,](https://football-data.co.uk/brazil.php) acesso em 06/2024.
6. Saraiva, Thiago [homepage na Internet]. Repositório do trabalho. Disponível em: <https://github.com/thg88/palpites_do_robo>[,](https://football-data.co.uk/brazil.php) acesso em 06/2024.