Modelagem Interpretável para Previsão de Partidas de Tênis com Regressão Simbólica

Nathan T. Braga
CEFET/RJ, Petrópolis, RJ

Resumo. Prever o resultado de partidas de tênis tem se mostrado uma tarefa de grande interesse em áreas como estatística esportiva, apostas e análise de desempenho, podendo auxiliar tanto torcedores quanto treinadores e analistas na tomada de decisões. No entanto, o tênis é um esporte dinâmico e complexo, no qual prever o resultado de uma partida representa um grande desafio devido à influência de diversas variáveis contextuais e relacionadas ao desempenho dos jogadores. Modelos preditivos tradicionais, como Artificial Neural Networks (ANN), Random Forests (RF) e Support Vector Machines (SVM), costumam funcionar como caixas-pretas, oferecendo pouca interpretabilidade sobre os fatores que influenciam suas decisões. Neste trabalho, é explorado o uso da regressão simbólica (Symbolic Regression) não apenas para prever os resultados de partidas de tênis, mas também para extrair insights interpretáveis sobre quais variáveis mais impactam a precisão das previsões. Através da geração de expressões matemáticas compreensíveis, a regressão simbólica permite identificar indicadores de desempenho relevantes, aliando poder preditivo à transparência do modelo.

Palavras-chave. Aprendizado de Máquina, Regressão Simbólica, Partidas de tênis, Previsão de vitória.

1 Introdução

O tênis é um dos esportes mais populares e acompanhados do mundo, tanto por seu apelo competitivo quanto pela complexidade estratégica envolvida nas partidas. Diferentemente de esportes coletivos, o desempenho individual no tênis depende de uma ampla gama de fatores físicos, técnicos, mentais e até mesmo emocionais. Esse conjunto de variáveis torna o esporte um terreno fértil para estudos analíticos e preditivos, despertando o interesse de pesquisadores das áreas de ciência de dados, inteligência artificial e estatística aplicada [1]. [2]

A crescente digitalização do esporte e o avanço das tecnologias de rastreamento e coleta de dados permitiram a criação de bancos de dados cada vez mais detalhados sobre partidas, jogadores e condições de jogo. Atualmente, é possível acessar informações que vão desde estatísticas básicas, como número de aces e duplas faltas, até dados mais refinados, como a velocidade média do saque, pontos ganhos em segundo serviço e eficiência em break points. Esse volume de dados proporciona uma base sólida para o desenvolvimento de modelos computacionais voltados à previsão de resultados [3].

Com o avanço da aprendizagem de máquina (machine learning), diferentes técnicas têm sido aplicadas na tentativa de prever o desfecho de partidas de tênis. Modelos como Artificial Neural Networks (ANN), Random Forests (RF) e Support Vector Machines (SVM) apresentam bons resultados preditivos, mas carecem de interpretabilidade, ou seja, apesar de preverem bem, não deixam claro "por que" ou "como" chegaram a uma determinada conclusão [4][5]. Essa falta de transparência pode ser um obstáculo tanto na análise de desempenho esportivo quanto na aplicação prática de estratégias baseadas em dados.

A interpretabilidade tem se tornado um tema central na ciência de dados, especialmente em áreas nas quais a confiança nos modelos é crucial. Em contextos esportivos, onde treinadores, atletas e analistas precisam compreender os fatores que influenciam a performance, utilizar modelos que funcionam como verdadeiras "caixas-pretas" pode ser contraproducente. Nesse cenário, métodos que conciliem desempenho preditivo com transparência são cada vez mais valorizados, especialmente quando a compreensão dos padrões é tão relevante quanto a própria previsão [6]

Diversos estudos recentes exploram o uso de técnicas de aprendizado de máquina para prever os resultados de partidas de tênis. [2] utilizaram modelos como Random Forest, SVM e Regressão Logística aplicados a um extenso conjunto de dados de partidas da ATP entre 2000 e 2016, alcançando acurácia de até 80%. O estudo identificou a força do saque como um dos principais fatores determinantes para a vitória. Já [7] propôs uma abordagem híbrida baseada em lógica fuzzy, redes neurais e uma equação de força, que combina diferentes fontes de informação, como o desempenho recente dos jogadores e a superfície da quadra, para gerar predições mais precisas. Esses trabalhos demonstram a relevância e a diversidade de métodos utilizados para modelar esse tipo de problema.

Neste trabalho, será utilizada a técnica de regressão simbólica para prever o resultado de partidas de tênis com base em diversos parâmetros estatísticos, utilizando um conjunto de dados disponível no Kaggle. A análise envolve tanto a construção de modelos preditivos quanto a avaliação de seu desempenho por meio de métricas de erro, como RMSE, MAE, MSE e R^2. Além disso, serão apresentados gráficos comparativos dessas métricas e uma análise interpretativa com base nas expressões geradas, destacando quais variáveis mais influenciam a previsão dos resultados.

2 Metodologia

2.1 Conjunto de dados

Neste estudo, foi utilizada uma base de dados disponível no Kaggle intitulada "Huge Tennis Database", que reúne informações detalhadas sobre partidas de tênis disputadas desde 1968. A base contém milhões de registros abrangendo diferentes categorias de jogo, incluindo partidas individuais, em duplas, eventos profissionais e amadores. Devido ao grande volume de dados e visando focar em informações mais recentes e relevantes, optou-se por utilizar somente os registros correspondentes ao ano de 2024, o que também contribui para a redução da dimensão do conjunto de dados e facilita a execução dos experimentos computacionais.

A base de dados está organizada no formato SQLite e estruturada em múltiplas tabelas interrelacionadas. Entre as principais tabelas disponíveis estão: players, que contém informações detalhadas dos jogadores como identificador único, nome, data de nascimento e mão dominante; matches, que registra cada partida com dados como data, local, tipo de superfície, rodadas, placar e identificação do vencedor; e rankings, que acompanha a evolução da posição dos jogadores ao longo do tempo.

Para a construção do modelo preditivo, optou-se por utilizar apenas os dados referentes a partidas individuais (singles), desconsiderando confrontos em duplas ou outros formatos. As variáveis selecionadas para análise incluem altura, idade, ranking, mão dominante (hand) e tipo de superfície da quadra (surface). Como o banco de dados original não possuía uma coluna explícita indicando a vitória ou derrota de um dos jogadores de forma binária, foi necessário realizar um pré-processamento adicional: as variáveis numéricas foram transformadas em diferenças entre o vencedor e o perdedor, enquanto as variáveis categóricas foram codificadas adequadamente. Com isso, foi criada uma nova coluna indicadora do resultado da partida, sinalizando com 1 se o jogador analisado venceu e com 0 caso contrário. A seguir, serão apresentados os possíveis valores assumidos pelas variáveis categóricas codificadas para a análise:

- **surface**: variável categórica representando o tipo de quadra. Foi codificada da seguinte forma:
  - 1 = quadra dura
  - 0 = saibro
- **diff_hand**: representa a diferença de dominância manual entre os jogadores. Os valores possíveis são:
  - 0 = ambos os jogadores possuem a mesma dominância (ex: destro vs destro ou canhoto vs canhoto)
  - 1 = o vencedor é destro e o perdedor é canhoto
  - -1 = o vencedor é canhoto e o perdedor é destro

A seguir, são descritas as variáveis numéricas e como foram tratadas no processo de preparação dos dados:

- **diff_rank**: diferença entre o ranking do vencedor e o do perdedor. Valores positivos indicam que o vencedor possuía um ranking pior (número maior), enquanto valores negativos indicam que o vencedor já estava mais bem ranqueado.
- **diff_ht**: diferença de altura entre os jogadores, medida em centímetros (cm). Um valor positivo indica que o vencedor era mais alto que o perdedor.
- **diff_age**: diferença de idade entre os jogadores, expressa em anos. Valores positivos indicam que o vencedor era mais velho que o perdedor, e negativos indicam o oposto.

Após o primeiro estágio de pré-processamento, que incluiu a filtragem dos dados e a construção das variáveis diferenciais, foi realizada a remoção de outliers visando eliminar valores extremos que distorcessem o desempenho do modelo. Em seguida, as instâncias do conjunto de dados foram embaralhadas aleatoriamente para evitar qualquer viés relacionado à ordem temporal ou agrupamento de partidas similares. Por fim, o dataset foi dividido em três subconjuntos: 70% para treinamento, 10% para validação e 20% para teste, garantindo uma separação adequada para o ajuste e a avaliação do modelo preditivo com base em dados não vistos.

2.2 A Regressão Simbólica

A Regressão Simbólica (SR) é uma técnica poderosa de aprendizado de máquina que busca descobrir expressões interpretáveis capazes de explicar as relações subjacentes em um determinado conjunto de dados. Diferentemente dos modelos de regressão tradicionais, que dependem de formas funcionais pré-definidas (como linear, polinomial ou exponencial), a SR realiza simultaneamente a busca pela estrutura e pelos parâmetros da expressão matemática que melhor se ajusta aos dados. Essa busca dual torna a SR uma abordagem de modelagem particularmente flexível e expressiva, especialmente em domínios onde a forma funcional da relação entre as variáveis é desconhecida ou complexa[8].

Uma das principais vantagens da regressão simbólica está em sua interpretabilidade. Modelos de caixa-preta, como redes neurais ou métodos de comitê (ensemble), geralmente não revelam a lógica interna por trás de suas previsões, enquanto a regressão simbólica produz equações explícitas que podem fornecer insights científicos e favorecer a intuição física sobre os fenômenos modelados [9].

Na regressão simbólica, as equações geradas são comumente representadas na forma de árvores de expressão, nas quais os nós internos correspondem a operadores matemáticos (como soma,

multiplicação ou funções não lineares), e os nós terminais representam variáveis de entrada ou constantes. Essa estrutura em árvore permite a construção de expressões matemáticas de forma flexível e modular, facilitando tanto a busca por modelos quanto sua posterior interpretação [10].

A regressão simbólica geralmente se baseia em algoritmos evolutivos para realizar a busca pelas melhores expressões matemáticas, sendo os operadores de seleção, crossover e mutação componentes fundamentais desse processo.

O operador de seleção visa escolher, na população atual de equações candidatas, aquelas que apresentarão maior aptidão (fitness) com base em critérios como erro de predição e complexidade da expressão. Métodos comuns de seleção incluem torneio, roleta e seleção elitista, e garantem que as expressões mais promissoras tenham maior chance de gerar descendentes [11].

O crossover, ou recombinação, atua combinando partes de duas árvores de expressão diferentes para gerar uma nova expressão, por exemplo, trocando subárvores entre indivíduos, promovendo diversidade e permitindo a criação de soluções mais complexas a partir de componentes já existentes [12].

Já a mutação consiste em alterações aleatórias em uma única árvore de expressão, como a substituição de um operador por outro, a modificação de uma constante ou a troca de uma subárvore inteira. A mutação é essencial para explorar novas regiões do espaço de busca e evitar o risco de convergência prematura para mínimos locais [13]. Esses operadores trabalham em conjunto ao longo de diversas gerações, permitindo que o algoritmo evolua expressões cada vez mais precisas e interpretáveis para representar as relações presentes nos dados.

Neste trabalho, usou-se o PySR [14] para realizar a regressão simbólica devido à sua capacidade de descobrir, eficientemente, expressões matemáticas interpretáveis a partir dos dados. O PySR é uma biblioteca de código aberto que combina o poder expressivo da regressão simbólica com a robustez de algoritmos evolutivos, oferecendo uma estrutura prática para a descoberta de equações em contextos científicos e de engenharia.

O algoritmo central empregado pelo PySR segue o paradigma evoluir–simplificar–otimizar. As expressões candidatas são evoluídas por meio de uma estratégia de programação genética com múltiplas populações, incorporando operadores padrão como crossover e mutação, além de elitismo e seleção por torneio. A cada geração, são aplicadas rotinas de simplificação simbólica para reduzir redundâncias nas expressões, seguidas de ajustes finos nos parâmetros para melhorar a precisão numérica. O sistema também inclui um mecanismo adaptativo de parcimônia, que penaliza modelos excessivamente complexos durante o processo de busca, promovendo assim a interpretabilidade e a capacidade de generalização das soluções encontradas.

2.3 Configuração do Modelo

Uma das principais vantagens do uso do PySR é o alto grau de configurabilidade oferecido por meio de sua interface PySRRegressor. Como destacado por [14], a biblioteca foi projetada especificamente para a descoberta científica, permitindo que pesquisadores ajustem praticamente todas as etapas do pipeline de regressão simbólica. É possível definir conjuntos personalizados de operadores unários e binários, modificar as heurísticas da busca evolutiva, impor restrições específicas do problema e até fornecer funções de perda personalizadas. Esse controle abrangente permite alinhar o espaço de busca do modelo ao conhecimento prévio do domínio e às particularidades dos dados, facilitando assim a extração de expressões concisas e fisicamente interpretáveis, em contraste com modelos de caixa-preta opacos.

Vários parâmetros precisam ser definidos para guiar a busca pela melhor equação. Para isso, definiram-se os parâmetros em busca da melhor equação com menor complexidade, respeitando os limites físicos do hardware utilizado para treinar o modelo. Tais parâmetros principais são:

- **populations**: Especifica o número de subpopulações isoladas (ou "ilhas") que evoluem em

paralelo. Cada ilha passa por um processo evolutivo independente, com migração ocasional de indivíduos entre elas, promovendo diversidade e reduz o risco de convergência prematura.

- **population_size**: Define o número máximo de indivíduos (isto é, equações candidatas) mantidos em cada ilha a qualquer momento. Populações maiores aumentam a cobertura do espaço de busca, mas também elevam o custo computacional.
- **maxsize**: Controla a complexidade máxima de uma expressão individual, limitando o número total de nós em sua árvore de expressão. Esse parâmetro atua como uma restrição direta sobre a interpretabilidade dos modelos gerados, já que árvores menores tendem a resultar em fórmulas mais concisas e compreensíveis.
- **niterations**: Define quantas vezes cada população passará por um ciclo evolutivo completo. Um número maior de iterações permite maior refinamento das expressões, mas também demanda mais tempo de processamento.

Além disso, o PySR permite definir explicitamente os operadores unários e binários que serão utilizados durante a construção das expressões matemáticas. Essa flexibilidade é fundamental para alinhar o modelo às características do problema estudado, restringindo a busca a operações matematicamente válidas ou fisicamente interpretáveis. A Tabela 1 apresenta os operadores utilizados neste trabalho.

Tipo	**Operadores**
Binary	$+$, $\times$, $-$, $/$, $\hat{}$
Unary	sin, cos, exp, log, sinh, cosh, erf

Tabela 1: Operadores definidos para o modelo

A Tabela 2 evidencia os hiperparâmetros que foram definidos para o treinamento do modelo.

Hiperparâmetro	**Valor**
Populations	100
Population_size	200
Maxsize	30
Niterations	10^4

Tabela 2: Hiperparâmetros utilizados no modelo

3 Resultados

A regressão simbólica seleciona a melhor equação com base em uma análise combinada de acurácia e complexidade. A acurácia é determinada pela capacidade do modelo de prever corretamente os valores de saída, enquanto a complexidade é medida pela estrutura da árvore simbólica, onde cada nó representa uma unidade de complexidade. Durante o processo de aprendizagem, a regressão simbólica busca equilibrar esses dois fatores, escolhendo a equação que oferece o melhor compromisso entre precisão na previsão e simplicidade do modelo, garantindo que ele seja eficiente, sem perder a capacidade de generalizar os dados.

Antes de analisar o desempenho do modelo nos dados de teste, precisa-se escolher uma equação candidata utilizando os dados de validação. Esse procedimento é fundamental para garantir que o modelo não esteja apenas decorando os dados de treino (overfitting), mas sim generalizando de maneira eficaz para dados não vistos. Durante o treinamento, o modelo é ajustado com base nos dados de treino, e a validação serve como uma etapa intermediária para evitar que o modelo se ajuste excessivamente a esses dados. A equação (1) foi a equação com melhor *score* nos dados de validação:

$$y = sin(erf(diff\_rank) + 1.2628733e7) \tag{1}$$

Para avaliar o desempenho dessa equação candidata, foi realizado o cálculo da acurácia do modelo utilizando a fórmula clássica para avaliação de classificadores binários:

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

onde TP (True Positives) representa o número de acertos em que a previsão foi positiva e correta, TN (True Negatives) é o número de acertos em que a previsão foi negativa e correta, FP (False Positives) são os casos em que a previsão foi positiva, mas a classe real era negativa, e FN (False Negatives) é o número de vezes que o modelo previu negativamente, mas a classe real era positiva. Utilizou-se, também, um limiar em 0.5 que vai transformar a probabilidade de vitória em um binário, com 0 sendo uma provável derrota e 1 sendo uma provável vitória.

Ao analisar a melhor equação selecionada pela regressão simbólica, observou-se que ela possui uma complexidade de 5, o que indica que a equação é relativamente simples em termos de número de operações. A acurácia alcançada nos dados de validação foi de 0.595, sugerindo um desempenho razoável na previsão da vitória do jogador. Além disso, a equação selecionada utilizou apenas uma variável independente, destacando a escolha da regressão simbólica por uma abordagem mais simples. Ao aplicar a mesma equação nos dados de teste, a acurácia foi de 0.628, superando o desempenho obtido nos dados de validação. Esse aumento na acurácia pode indicar que o modelo, embora simples, conseguiu uma generalização melhor para os dados não vistos durante o treinamento.

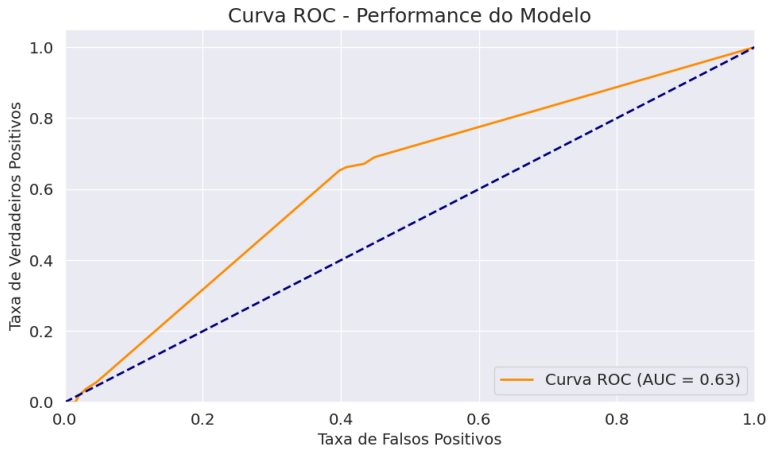

Figura 1: Curva ROC

Ademais, foi realizada a análise da curva ROC (Receiver Operating Characteristic), que é uma ferramenta gráfica usada para avaliar a performance de modelos de classificação binária. Ela plota a

taxa de verdadeiros positivos (TPR) contra a taxa de falsos positivos (FPR) para diferentes limiares de decisão, proporcionando uma visão clara da capacidade do modelo em distinguir entre as classes. A curva ROC resultante pode ser vista na Figura 1, onde é possível observar o desempenho do modelo em termos de sua capacidade discriminativa.

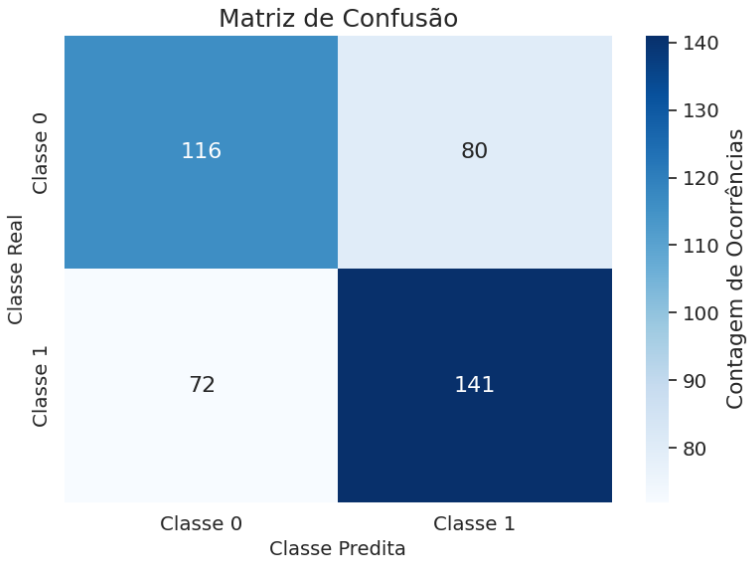

Figura 2: Matriz de confusão

Por fim, foi gerada a matriz de confusão, que permite visualizar a performance do modelo de classificação ao comparar as previsões feitas com os valores reais. A matriz apresenta o número de acertos e erros nas diferentes classes, permitindo uma análise detalhada de como o modelo classifica os dados. A matriz de confusão gerada para este modelo está apresentada na Figura 2, evidenciando o número de verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos, fundamentais para uma avaliação mais precisa do desempenho do modelo.

4 Conclusões

Neste trabalho, foi investigada a aplicação da regressão simbólica na previsão da vitória de jogadores de tênis, um problema desafiador e altamente complexo. Dada a complexidade intrínseca do problema de prever a vitória de um jogador de tênis, a regressão simbólica optou por selecionar uma equação mais simples, pois evidenciou-se que um aumento substancial na complexidade do modelo não resultaria em ganhos significativos de acurácia. A equação gerada, apesar de sua simplicidade, obteve desempenho considerável tanto nos dados de validação quanto nos dados de teste, com acurácias de 0.595 e 0.628, respectivamente.

O modelo resultante utilizou apenas uma variável independente, conseguindo equilibrar de forma eficaz acurácia e complexidade, evitando overfitting. Isso reforça a ideia de que, em certos casos, a escolha de um modelo mais simples pode ser mais eficaz do que aumentar a complexidade sem um ganho proporcional de performance. A análise da matriz de confusão e da curva ROC também proporcionou uma visão detalhada do comportamento do modelo, permitindo compreender a taxa de acerto e os erros cometidos em cada classe.

Referências

[1] Machar Reid, Stuart Morgan e David Whiteside. "Matchplay characteristics of Grand Slam tennis: Implications for training and conditioning". Em: **Journal of Sports Sciences** 34.19 (2016), pp. 1791–1798.

[2] Zijian Gao e Amanda Kowalczyk. "Random forest model identifies serve strength as a key predictor of tennis match outcome". Em: **Journal of Sports Analytics** 7.4 (2021), pp. 255–262. DOI: 10.3233/JSA-200515. eprint: https://doi.org/10.3233/JSA-200515. URL: https://doi.org/10.3233/JSA-200515.

[3] Stephanie A. Kovalchik. "Searching for the GOAT of tennis win prediction". Em: **Journal of Quantitative Analysis in Sports** 16.1 (2020), pp. 1–15.

[4] Leo Breiman. "Random forests". Em: **Machine learning** 45.1 (2001), pp. 5–32.

[5] Trevor Hastie, Robert Tibshirani e Jerome Friedman. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. 2ª ed. New York: Springer, 2009.

[6] Finale Doshi-Velez e Been Kim. **Towards a rigorous science of interpretable machine learning**. arXiv preprint arXiv:1702.08608. 2017. arXiv: 1702.08608 [cs.LG].

[7] Mateus de Araujo Fernandes. "Using Soft Computing Techniques for Prediction of Winners in Tennis Matches". Em: **Machine Learning Research** 2.3 (2017), pp. 86–98. DOI: 10.11648/j.mlr.20170203.12. eprint: https://article.sciencepublishinggroup.com/pdf/10.11648.j.mlr.20170203.12. URL: https://doi.org/10.11648/j.mlr.20170203.12.

[8] Dimitrios Angelis, Filippos Sofos e Theodoros E Karakasidis. "Artificial intelligence in physical sciences: Symbolic regression trends and perspectives". Em: **Archives of Computational Methods in Engineering** 30.6 (2023), pp. 3845–3865.

[9] Nour Makke e Sanjay Chawla. "Interpretable scientific discovery with symbolic regression: a review". Em: **Artificial Intelligence Review** 57.1 (jan. de 2024). ISSN: 1573-7462. DOI: 10.1007/s10462-023-10622-0. URL: http://dx.doi.org/10.1007/s10462-023-10622-0.

[10] Candida Ferreira. "Gene expression programming: a new adaptive algorithm for solving problems". Em: **arXiv preprint cs/0102027** (2001).

[11] Mark E Kotanchek, Ekaterina Y Vladislavleva e Guido F Smits. "Symbolic regression via genetic programming as a discovery engine: Insights on outliers and prototypes". Em: **Genetic Programming Theory and Practice VII**. Springer, 2009, pp. 55–72.

[12] Laura S de Assis, Jurair R de P Junior, Luis Tarrataca e Diego B Haddad. "Efficient Volterra systems identification using hierarchical genetic algorithms". Em: **Applied Soft Computing** 85 (2019), p. 105745.

[13] Yongqiang Zhang, Huifang Cheng et al. "Improved genetic programming algorithm applied to symbolic regression and software reliability modeling". Em: **Journal of Software Engineering and Applications** 2.05 (2009), p. 354.

[14] Miles Cranmer. "Interpretable machine learning for science with PySR and SymbolicRegression.jl". Em: **arXiv preprint arXiv:2305.01582** (2023).

A Equações geradas pela Regressão Simbólica

Equations

$$y = 0.493$$

$$y = \cos\left(1.00^{diff_{rank}}\right)$$

$$y = \sin\left(\operatorname{erf}\left(diff_{rank}\right) + 1.26 \cdot 10^{7}\right)$$

$$y = 0.496 - \frac{0.162}{\operatorname{erf}\left(diff_{rank}\right)}$$

$$y = e^{-0.757 - \frac{0.339}{\operatorname{erf}\left(diff_{rank}\right)}}$$

$$y = \cos\left(\cosh\left(surface\right)0.157 + \operatorname{erf}\left(diff_{rank}\right)\right)$$

$$y = 0.496 - \frac{0.110}{\cos\left(surface\right)\frac{1}{\operatorname{erf}\left(diff_{rank}\right)}}$$

$$y = \sin\left(\sin\left(\operatorname{erf}\left(1.05^{diff_{rank}} diff_{rank}\right) - 1.04 \cdot 10^{7}\right)\right)$$

$$y = 0.497 - \frac{0.131}{(-1)\operatorname{erf}\left(\frac{0.396}{\operatorname{erf}\left(diff_{rank}(-0.00661)\right)}\right)}$$

$$y = 0.496 - \frac{0.164}{\operatorname{erf}\left(diff_{rank} - \sin\left(\sinh\left(diff_{ht}\right)\right)2.88\right)}$$

$$y = \sin\left(-\frac{0.172}{\operatorname{erf}\left(diff_{rank} + \sin\left(\sinh\left(diff_{ht}\right)\right)(-2.66)\right)}\right) + 0.496$$

$$y = 0.496 - \frac{0.172}{\operatorname{erf}\left(diff_{rank} - \sin\left(\sinh\left(diff_{age}(-1.28)\right)\right)2.39\right)}$$

$$y = 0.495 - \frac{0.110}{\cos\left(surface\right)\operatorname{erf}\left(diff_{rank} - \sin\left(\sinh\left(diff_{ht}\right)\right)2.91\right)}$$

$$y = \sin\left(-\frac{0.115}{\cos\left(surface\right)\operatorname{erf}\left(diff_{rank} - \sin\left(\sinh\left(diff_{ht}\right)\right)2.61\right)}\right) + 0.496$$

$$y = \sin\left(-\frac{0.178}{\operatorname{erf}\left(diff_{rank} + \sin\left(\frac{diff_{age}}{-0.0544} - \sinh\left(diff_{ht}\right)\right)4.38\right)}\right) + 0.498$$

$$y = \sin\left((-1)\left(-0.0206\right) - \frac{0.178}{\operatorname{erf}\left(diff_{rank} + \sin\left(\frac{diff_{age}}{-0.0544} - \sinh\left(diff_{ht}\right)\right)4.38\right)}\right) + 0.478$$

$$y = \sin\left(-\frac{0.118}{\cos\left(surface\right)\operatorname{erf}\left(diff_{rank} + \sin\left(\frac{diff_{age}}{-0.0544} - \sinh\left(diff_{ht}\right)\right)4.24\right)}\right) + 0.497$$

Equations

$$y = 0.498 - \frac{0.151}{(-1)\operatorname{erf}\left(\frac{0.398}{\operatorname{erf}\left(\left(diff_{rank}+\sin\left(\frac{diff_{age}}{-0.0827}-\sinh\left(diff_{ht}\right)\right)4.17\right)(-0.00515)\right)}\right)}$$

$$y = 0.493 - \frac{0.151}{\operatorname{erf}\left(0.0520-\frac{0.411}{\operatorname{erf}\left(\left(diff_{rank}+\sin\left(\frac{diff_{age}}{-0.0827}-\sinh\left(diff_{ht}\right)\right)4.17\right)(-0.00545)\right)}\right)}$$

$$y = \cosh\left(surface\right)(-0.118)\frac{1}{(-1)\operatorname{erf}\left(\frac{0.423}{\operatorname{erf}\left(\left(diff_{rank}+\sin\left(\frac{diff_{age}}{-0.0827}-\sinh\left(diff_{ht}\right)\right)4.17\right)(-0.00552)\right)}\right)} + 0.498$$

$$y = \cosh\left(\frac{1.84}{diff_{rank}}\right)(-0.139)\frac{1}{(-1)\operatorname{erf}\left(\frac{0.364}{\operatorname{erf}\left(\left(diff_{rank}+\sin\left(\frac{diff_{age}}{-0.0827}-\sinh\left(diff_{ht}\right)\right)4.17\right)(-0.00535)\right)}\right)} + 0.498$$

$$y = \frac{\sin\left(\cosh\left(-\frac{3.75}{diff_{rank}}\right)(-0.127)\right)}{(-1)\operatorname{erf}\left(\frac{0.331}{\operatorname{erf}\left(\left(diff_{rank}+\sin\left(\frac{diff_{age}}{-0.0827}-\sinh\left(diff_{ht}\right)\right)4.17\right)(-0.00560)\right)}\right)} + 0.498$$

$$y = 0.499 - \frac{0.130}{\cos\left(\sinh\left(\sin\left(\frac{4.65}{diff_{rank}}\right)\right)\right)\left(-\operatorname{erf}\left(\frac{0.347}{\operatorname{erf}\left(\left(diff_{rank}+\sin\left(\frac{diff_{age}}{-0.0827}-\sinh\left(diff_{ht}\right)\right)4.17\right)(-0.00564)\right)}\right)\right)}$$

$$y = \cosh\left(surface\right)\cosh\left(\frac{1.72}{diff_{rank}}\right)(-0.109)\frac{1}{(-1)\operatorname{erf}\left(\frac{0.386}{\operatorname{erf}\left(\left(diff_{rank}+\sin\left(\frac{diff_{age}}{-0.0827}-\sinh\left(diff_{ht}\right)\right)4.17\right)(-0.00568)\right)}\right)} +$$

$$0.498$$

$$y = 0.498 - \frac{0.102}{\cos\left(\sinh\left(\operatorname{erf}\left(\frac{diff_{age}}{diff_{rank}}+surface\right)\right)\right)\left(-\operatorname{erf}\left(\frac{0.414}{\operatorname{erf}\left(\left(diff_{rank}+\sin\left(\frac{diff_{age}}{-0.0827}-\sinh\left(diff_{ht}\right)\right)4.17\right)(-0.00614)\right)}\right)\right)}$$