Comparative Analysis between Genetic Programming and Machine Learning Algorithms in Forecasting Financial Trends*

Marcos V. R. Pedroza[1], **Carlos A. Silva**[1]

[1]Departamento de Informática – Instituto Federal de Minas Gerais (IFMG)
CEP 34.590-390 – Sabará, MG – Brasil

marcospdrza@gmail.com, carlos.silva@ifmg.edu.br

Abstract. *This study presents a comparative analysis of Genetic Programming (GP) and five machine learning (ML) algorithms, namely Support Vector Machines (SVM), AdaBoost, XGBoost, Long Short-Term Memory (LSTM), and Deep Neural Networks (DNN), in the task of financial trend forecasting. We use historical daily data from the NASDAQ, S&P 500, and Nikkei 225 indices, covering the period from January 2015 to January 2025. Model performance is evaluated using Sharpe and Sortino Ratios, capturing both accuracy and risk-adjusted return. Results show that GP exhibits greater stability in Asian markets, while LSTM and XGBoost achieve better performance in North American markets.*

Resumo. *Este estudo apresenta uma análise comparativa entre a Programação Genética (GP) e cinco algoritmos de aprendizado de máquina (SVM, AdaBoost, XGBoost, LSTM e DNN) aplicados à previsão de tendências financeiras. Os dados utilizados compreendem séries históricas diárias dos índices NASDAQ, S&P 500 e Nikkei 225, no período de janeiro de 2015 a janeiro de 2025. A avaliação do desempenho foi realizada com base nas métricas Sharpe Ratio e Sortino Ratio, considerando tanto o retorno quanto o risco associado. Os resultados indicam que a GP demonstrou maior estabilidade nos mercados asiáticos, enquanto os modelos LSTM e XGBoost se destacaram por seus desempenhos superiores em mercados norte-americanos.*

1. Introdução

A previsão financeira desempenha um papel estratégico no suporte à tomada de decisões de investimento, com o objetivo de maximizar retornos e mitigar riscos em ambientes de alta volatilidade. Diversas pesquisas vêm explorando modelos preditivos capazes de superar *benchmarks* de mercado sob determinadas condições e janelas temporais específicas. Tais modelos diferem amplamente em termos de complexidade, interpretabilidade e aplicabilidade, sendo que a escolha do algoritmo ideal depende do tipo de ativo, horizonte temporal e cenário de mercado considerados.

A análise técnica, fundamentada em dados históricos, enfrenta desafios significativos na identificação de padrões robustos e generalizáveis. Nos últimos anos, algoritmos de aprendizado de máquina (*Machine Learning*, ML) têm se destacado por sua capacidade de extrair padrões complexos de séries temporais financeiras, conforme evidenciado por

*Artigo aceito para publicação no XXII Encontro Nacional de Inteligência Artificial e Computacional, Fortaleza, CE, set. 2025.

[Atsalakis et al. 2019], que compara ML com abordagens tradicionais. Nesse cenário, a programação genética (*Genetic Programming*, GP) surge como uma abordagem baseada em princípios evolutivos, com potencial para construir modelos preditivos adaptativos e de difícil formulação analítica.

Embora existam comparações pontuais entre GP e técnicas de aprendizado de máquina, este estudo propõe uma análise mais abrangente, incorporando múltiplas abordagens de ML e considerando diferentes mercados e períodos históricos. Especificamente, este estudo avalia a GP em relação a cinco algoritmos de ML: Support Vector Machines (SVM), AdaBoost, XGBoost, Long Short-Term Memory (LSTM) e Deep Neural Networks (DNN). Estudos anteriores, como os de [Krauss et al. 2017] sobre SVM e DNN aplicadas ao índice S&P 500, de [Fischer and Krauss 2018] sobre LSTM, e de [Long et al. 2020], que compararam GP a diferentes algoritmos de ML, oferecem uma base relevante. Este trabalho busca complementar essas pesquisas ao comparar o desempenho dos referidos algoritmos com a GP em diferentes mercados e horizontes temporais.

Os experimentos realizados utilizaram conjuntos de dados heterogêneos e exploraram o impacto do período de treinamento de dez anos sobre o desempenho dos algoritmos. Ao aplicar essas metodologias em diversos mercados acionários, buscou-se identificar os modelos com maior capacidade de generalização e consistência, de modo a subsidiar decisões informadas por dados sob distintos perfis de risco. Optou-se pelo uso do NASDAQ e do S&P 500, representando o mercado norte-americano, e do Nikkei 225, representando o mercado asiático, devido à relevância histórica e à diversidade desses índices, que permitem avaliar os modelos em contextos distintos de maturidade e volatilidade.

Assim sendo, este artigo está estruturado da seguinte forma: a seção 2 apresenta a revisão da literatura, com foco em fundamentos de análise técnica e aplicações recentes de GP na previsão financeira. A seção 3 descreve a metodologia adotada, incluindo a preparação dos dados, os indicadores técnicos utilizados, os algoritmos avaliados e a configuração da GP. A seção 4 discute os resultados obtidos por cada modelo nos diferentes mercados, utilizando métricas de erro e de retorno ajustado ao risco. Por fim, a seção 5 apresenta as conclusões do estudo, suas limitações e sugestões para pesquisas futuras.

2. Trabalhos Correlatos

A previsão financeira com uso de ML tem sido objeto de recentes pesquisas, refletindo o dinamismo dessa área. O estudo de [Atsalakis et al. 2019] demonstrou que modelos como SVM e DNN podem superar abordagens clássicas da análise baseada em indicadores históricos em determinados mercados. Contudo, essas soluções ainda enfrentam limitações significativas, como alta sensibilidade aos dados e baixa robustez em cenários voláteis, o que evidencia a demanda por métodos mais generalizáveis.

Os autores de [Krauss et al. 2017] investigaram o desempenho de SVM e DNN no contexto do S&P 500, destacando que, apesar dos bons resultados obtidos, a eficácia desses modelos é altamente dependente dos hiperparâmetros e da janela temporal de treinamento, o que pode comprometer sua reprodutibilidade em contextos distintos. Complementando essa análise, [Fischer and Krauss 2018] analisaram o uso de redes do tipo LSTM para previsão de tendências em séries temporais financeiras, com foco no índice S&P 500. Seus resultados indicaram que, em ambientes de alta volatilidade, modelos LSTM superam técnicas tradicionais por conseguirem capturar dependências temporais

complexas, tornando-se uma alternativa promissora para cenários com ruído e sazonalidade.

No campo da GP, [Chen et al. 2021] demonstraram sua aplicabilidade na construção de estratégias de negociação adaptativas. Embora a GP requeira maior capacidade computacional, seus modelos se mostram eficientes em contextos com elevada variabilidade e difícil modelagem analítica. O trabalho de [Shen et al. 2020], por sua vez, explorou modelos híbridos combinando GP com algoritmos de ML, destacando que essa integração permite não apenas identificar padrões complexos, mas também refinar previsões, sendo especialmente útil em mercados emergentes, onde a volatilidade tende a ser maior.

Embora avanços relevantes tenham sido apresentados por [Long et al. 2020], que compararam GP com nove algoritmos de ML em tarefas de previsão financeira, ainda permanecem lacunas quanto à análise em múltiplos mercados e à avaliação comparativa em contextos históricos distintos. Diferentemente desse estudo, o presente trabalho concentra-se em cinco algoritmos amplamente utilizados — SVM, *AdaBoost*, *XGBoost*, LSTM e DNN — para uma comparação direta com GP. O objetivo é compreender melhor os pontos fortes e limitações da GP em relação a modelos supervisionados, fornecendo evidências empíricas sobre sua aplicabilidade e robustez em ambientes complexos.

3. Metodologia

Nesta seção, descreve-se a metodologia empregada na previsão de tendências financeiras, elaborada a partir do trabalho de [Long et al. 2020], que serviu de inspiração para a abordagem adotada.

3.1. Coleta e Preparação dos Dados

A base de dados foi construída com informações históricas de três índices financeiros amplamente reconhecidos como referências de mercado: NASDAQ, S&P 500 e Nikkei 225 [Britannica Money 2025]. Os dados foram obtidos por meio da biblioteca *yfinance*, em frequência diária, abrangendo o período de janeiro de 2015 a janeiro de 2025. Esse intervalo possibilita capturar variações sazonais, ciclos econômicos e eventos globais com impacto direto sobre os mercados financeiros.

A coleta resultou em dataframes com *multi-index*, uma estrutura de indexação hierárquica que permite múltiplos níveis de rótulos para linhas ou colunas. Embora seja útil para organizar dados complexos, o *multi-index* dificulta operações simples de seleção e manipulação. Por esse motivo, realizou-se uma etapa inicial de padronização, na qual os índices compostos foram removidos e as colunas renomeadas, a fim de facilitar o acesso e o tratamento subsequente. Foram extraídas as seguintes colunas para cada índice: data, preço de abertura (*Open*), preço máximo (*High*), preço mínimo (*Low*), preço de fechamento (*Close*) e volume negociado (*Volume*).

Após a padronização, foi realizada a interpolação linear de dados temporais ausentes, com foco especial na coluna de datas, respeitando a ordem temporal. Para lidar com valores ausentes nas colunas numéricas, foi utilizado o método *KNN Imputer* [Troyanskaya et al. 2001], uma abordagem baseada em similaridade que preserva padrões não lineares e características sazonais do mercado.

3.2. Indicadores Utilizados

Para enriquecer o conjunto de dados e ampliar a capacidade preditiva dos modelos, foram selecionados indicadores técnicos agrupados em quatro categorias principais, referentes a diferentes aspectos do mercado financeiro, conforme analisado por [Mostafavi 2025]. Cada grupo reflete aspectos distintos do comportamento do mercado, permitindo uma análise mais abrangente e detalhada das séries temporais financeiras. Os indicadores estão listados na Tabela 1.

Tabela 1. Categorias de indicadores técnicos utilizadas neste estudo com suas fórmulas.

Categoria	**Indicador**	**Fórmula**
Trend	Exponential Moving Average	$EMA_t = \alpha P_t + (1-\alpha)EMA_{t-1}, \alpha = \frac{2}{N+1}$
	Parabolic Stop and Reverse	$SAR_{t+1} = SAR_t + AF \cdot (EP - SAR_t)$
	Moving Average	$MA_t = \frac{1}{N}\sum_{i=0}^{N-1} P_{t-i}$
Market Strength	Moving Average Convergence Divergence	$MACD = EMA_{12} - EMA_{26}$
	On-Balance Volume	$OBV_t = OBV_{t-1} + \begin{cases} +Volume_t, & P_t > P_{t-1} \\ -Volume_t, & P_t < P_{t-1} \\ 0, & P_t = P_{t-1} \end{cases}$
	Accumulation/Distribution	$A/D_t = A/D_{t-1} + \frac{(P_t - Low_t) - (High_t - P_t)}{High_t - Low_t} \cdot Volume_t$
Momentum	Relative Strength Index	$RSI = 100 - \frac{100}{1+RS}, RS = \frac{\text{Avg Gain}}{\text{Avg Loss}}$
	Stochastic Oscillator	$\%K = \frac{P_t - Low_N}{High_N - Low_N} \cdot 100$
	Commodity Channel Index	$CCI = \frac{P_t - SMA(P_t)}{0.015 \cdot \text{Mean Deviation}}$
Volatility	Average True Range	$ATR = \frac{1}{N}\sum_{i=1}^{N} TR_i, TR = \max(High - Low, \lvert High - Close_{prev}\rvert, \lvert Low - Close_{prev}\rvert)$
	Bollinger Bands	$BB_{upper} = MA + K\sigma, BB_{lower} = MA - K\sigma$
	Daily Percentage Change	$\Delta P_t = \frac{P_t - P_{t-1}}{P_{t-1}} \cdot 100$

Os indicadores de tendência (*Trend Features*) são fundamentais para capturar o comportamento direcional do mercado ao longo do tempo. Eles ajudam a identificar movimentos sustentados, sejam de alta ou baixa, permitindo que os modelos aprendam padrões consistentes e evitem ruídos de curto prazo. A inclusão dessas variáveis fornece uma base sólida para que os algoritmos compreendam a dinâmica geral dos preços, favorecendo a tomada de decisões alinhadas com as tendências predominantes do mercado.

Os indicadores de força de mercado (*Market Strength Features*) avaliam o volume e a intensidade por trás dos movimentos de preço, oferecendo uma visão mais profunda sobre a robustez das tendências observadas. Ao considerar a interação entre preço e volume, essas métricas auxiliam na distinção entre movimentos genuínos e movimentos ocasionais ou manipulados. Sua incorporação amplia a capacidade dos modelos em identificar sinais confiáveis que precedem alterações significativas no comportamento do mercado.

Indicadores de *momentum* são essenciais para detectar a velocidade e a magnitude das mudanças no preço, revelando momentos em que o mercado pode estar sobrecomprado ou sobrevendido. Essa categoria captura a força dos movimentos recentes, oferecendo informações importantes para prever reversões ou confirmações de tendências. A integração dessas informações proporciona uma dimensão adicional para antecipar mudanças rápidas no comportamento do ativo, aumentando a eficácia das previsões em cenários voláteis.

Por fim, os indicadores de volatilidade refletem a instabilidade e o risco associado ao comportamento dos preços, sendo aspectos críticos para a avaliação financeira. Indicadores dessa categoria permitem reconhecer períodos de maior incerteza e flutuações abruptas, que podem impactar diretamente nas decisões de investimento. Considerar a volatilidade aprimora a capacidade preditiva e possibilita a construção de estratégias que equilibram retorno e risco, elemento central para uma análise financeira robusta.

Cada uma dessas variáveis foi calculada sobre as séries de preços de fechamento dos índices, formando uma matriz de características que aumentou significativamente o potencial preditivo dos algoritmos. Além disso, a escolha desses indicadores foi pautada na complementaridade entre eles: enquanto alguns capturam tendências de longo prazo, outros respondem rapidamente a mudanças bruscas de *momentum* ou volatilidade local.

3.3. Avaliação e Comparação dos Modelos

O objetivo é compreender como cada algoritmo se comporta em um contexto de decisão real, no qual o retorno ajustado ao risco é mais relevante do que a simples acurácia pontual. Para isso, foram aplicados o *Sharpe Ratio* e o *Sortino Ratio*, que medem o desempenho das previsões levando em conta tanto o retorno quanto a volatilidade [Investopedia 2024]. Esses indicadores permitem comparar os modelos de forma mais alinhada ao que um investidor de fato consideraria útil.

O *Sharpe Ratio* mensura o retorno excedente sobre a taxa livre de risco por unidade de volatilidade total. Sua utilização permite comparar a eficiência de diferentes modelos em gerar retorno ajustado ao risco, especialmente em contextos onde há alta variabilidade nos resultados.

$$\frac{R_p - R_f}{\sigma_p} \tag{1}$$

O *Sortino Ratio* é uma versão mais refinada dessa métrica, pois considera apenas a volatilidade negativa, desconsiderando flutuações acima do retorno mínimo desejado. Isso o torna mais apropriado para avaliação de estratégias assimétricas, como as que buscam minimizar perdas sem necessariamente restringir ganhos.

$$\frac{R_p - R_f}{\sigma_d} \tag{2}$$

3.4. Pré-processamento Final e Otimização dos Modelos

Antes da implementação dos algoritmos, todas as linhas com valores nulos remanescentes foram removidas, sobretudo aquelas resultantes de indicadores técnicos que exigem janelas temporais maiores para gerar observações válidas iniciais. O resultado foi uma

base de dados completamente limpa, sem valores faltantes, garantindo integridade para os experimentos subsequentes.

Antes da implementação dos algoritmos, todas as linhas com valores nulos remanescentes foram removidas, sobretudo aquelas resultantes de indicadores técnicos que exigem janelas temporais maiores para gerar observações válidas iniciais. O resultado foi uma base de dados completamente limpa, sem valores faltantes, garantindo integridade para os experimentos subsequentes.

Para o treinamento dos modelos de aprendizado de máquina, foram utilizados como atributos os indicadores técnicos agrupados em quatro categorias principais: momentum, volatilidade, força e tendência, conforme analisado por Mostafavi [**?**]. O valor alvo (target) considerado foi o retorno futuro do ativo no próximo período de negociação, permitindo que os modelos aprendessem a prever a direção e magnitude das variações de preço.

Cada algoritmo foi implementado com ajustes específicos voltados à sua natureza e à prevenção de *overfitting*. O treinamento utilizou 80% dos dados, enquanto 20% foram reservados para teste, e foi aplicada validação cruzada com *TimeSeriesSplit* para respeitar a ordem temporal e garantir uma avaliação robusta do desempenho. A Tabela 2 resume as estratégias de aprimoramento de cada modelo e suas fundamentações técnicas.

Tabela 2. Aprimoramentos aplicados aos algoritmos.

Algoritmo	Aprimoramento	Fundamentação Técnica
XGBoost	Emprego de *early stopping rounds* com conjunto de validação temporal	Mitiga o sobreajuste ao interromper o treinamento com base no desempenho de validação, respeitando a ordem cronológica dos dados.
AdaBoost	Configuração do estimador base com profundidade limitada (*max depth* = 1)	Reduz a complexidade individual dos classificadores fracos, favorecendo a generalização do *ensemble* e evitando sobreajuste estrutural.
DNN	Inserção de camadas de *dropout* e *batch normalization* entre camadas densas	Promove regularização durante o treinamento e estabilização estatística das ativações, resultando em maior robustez preditiva.
SVM	Normalização dos atributos com *StandardScaler* e otimização de hiperparâmetros (*C, gamma*) para o *kernel* RBF	Assegura convergência adequada do algoritmo e evita viés induzido por variáveis em escalas distintas, além de controlar a complexidade da fronteira de decisão.
LSTM	Arquitetura com empilhamento moderado de camadas e ativação de *return sequences* apenas na primeira camada	Limita a profundidade da recorrência para evitar memorizações espúrias e promover aprendizado mais eficiente de padrões temporais relevantes.

Os principais hiperparâmetros utilizados em cada modelo estão listados na Tabela 3. Ressalta-se que toda a análise foi conduzida de forma empírica, sem automação de testes.

Tabela 3. Hiperparâmetros utilizados em cada algoritmo.

Algoritmo	Parâmetro	Valor
SVM	C	0.1
	gamma	1
	kernel	rbf
	Normalização	StandardScaler
AdaBoost	Estimator	base_estimator
	n_estimators	50
	learning_rate	0.5
XGBoost	n_estimators	1000
	learning_rate	0.1
	eval_metric	rmse
LSTM	Primeira camada LSTM	50 unidades, return_sequences=True
	Segunda camada LSTM	50 unidades, return_sequences=False
	Dropout	0.2 em cada camada
	Dense saída	1 neurônio
	Otimizador	Adam
	Função de perda	Mean Squared Error (MSE)
DNN	Épocas	100
	Batch_size	32
	Validação	Early stopping
	Output	1 neurônio (regressão)

3.5. Programação Genética

Neste estudo, a GP foi utilizada para gerar estratégias de negociação financeira orientadas à maximização do retorno ajustado ao risco, em especial o *Sortino Ratio*.

A implementação foi realizada em *Python*, com uso da biblioteca `DEAP`, utilizando árvores de expressão como representação dos indivíduos da população evolutiva. Cada indivíduo é composto por primitivos e terminais, representados por variáveis derivadas dos indicadores técnicos e constantes efêmeras aleatórias. A Tabela 4 resume os principais componentes utilizados.

Tabela 4. Componentes principais da Programação Genética no DEAP.

Componente	Descrição
Conjunto de Primitivas	Inclui operações lógicas (`logical_and`, `logical_or`) e comparativas (`greater_than`, `less_than`), além de uma função de divisão protegida (`protectedDiv`).
Conjunto de Terminais	Inclui as variáveis de entrada $x_0, x_1, \ldots, x_{N-1}$, derivadas dos indicadores financeiros, e constantes efêmeras aleatórias geradas nos intervalos `[-1, 1]` e `[-100, 100]`.
Representação	Cada indivíduo é representado por uma árvore de expressão (`gp.PrimitiveTree`), associada a um tipo personalizado criado com `creator.Individual`. A aptidão é definida com `creator.FitnessMax`, visando a maximização do *Sortino Ratio*.

A configuração dos parâmetros foi definida com base em experimentação empírica, visando equilíbrio entre desempenho e tempo de execução. A Tabela 5 apresenta os valores utilizados.

Tabela 5. Parâmetros da Programação Genética.

Parâmetro	Valor
Tamanho da População (`POPULATION_SIZE`)	150
Número de Gerações (`N_GENERATIONS`)	35
Probabilidade de Cruzamento (`CXPB`)	0.8
Probabilidade de Mutação (`MUTPB`)	0.35
Torneio de Seleção (`Tournament`)	5

Os valores dos parâmetros foram definidos com base em experimentação empírica. Foram realizados testes preliminares com diferentes configurações de tamanho de população, número de gerações e taxas de mutação e cruzamento. Optou-se pelos valores apresentados por proporcionarem um bom equilíbrio entre desempenho e tempo de execução nos experimentos realizados. No entanto, não foi realizado um processo sistemático de otimização.

O processo evolutivo consistiu em três etapas principais: inicialização da população, execução do laço evolutivo com operadores de seleção, cruzamento e mutação (via `algorithms.eaSimple`), e armazenamento do melhor programa evoluído. A função de avaliação (`evaluate_program`) foi adaptada dinamicamente para cada índice de mercado (S&P 500, NASDAQ e Nikkei), garantindo que a evolução ocorresse no contexto de dados apropriado e permitindo a geração de estratégias específicas para cada mercado analisado.

A função de *fitness* foi projetada para quantificar a qualidade de cada estratégia evoluída, utilizando como métrica principal o *Sortino Ratio*, que penaliza apenas as perdas e, por isso, se aproxima mais do cenário real enfrentado pelos investidores em previsão financeira [Investopedia 2024]. Primeiramente, a árvore sintática do indivíduo é compilada com `toolbox.compile()`. Em seguida, essa função é aplicada linha a linha sobre o conjunto de treino, gerando sinais de negociação: valores superiores a 0.5 são interpretados como *compra* (1.0), e os demais como *manutenção* (0.0). Os retornos simulados são obtidos pela multiplicação do sinal pela variação diária do ativo.

O desempenho da estratégia é então avaliado com o *Sortino Ratio*, que penaliza apenas a volatilidade negativa — favorecendo estratégias que maximizam retorno com menor exposição ao risco de perda. Erros de execução, como divisão por zero, resultam automaticamente em penalizações de aptidão.

Essa abordagem permitiu a geração de estratégias personalizadas para cada mercado, refletindo as características específicas de volatilidade e comportamento de cada índice analisado.

4. Resultados e Discussão

Esta seção apresenta e interpreta os resultados obtidos a partir da validação cruzada dos modelos aplicados aos três índices financeiros analisados: S&P 500, Nikkei 225 e NASDAQ. Foram utilizadas quatro métricas: RMSE e MAE para quantificação do erro de previsão, além de *Sharpe Ratio* e *Sortino Ratio* para avaliação do retorno ajustado ao risco.

Desempenho no Índice S&P 500

A Figura 1 apresenta os resultados obtidos pelos algoritmos no índice S&P 500, comparando as métricas de erro (RMSE e MAE) e de risco-retorno (*Sharpe* e *Sortino Ratios*).

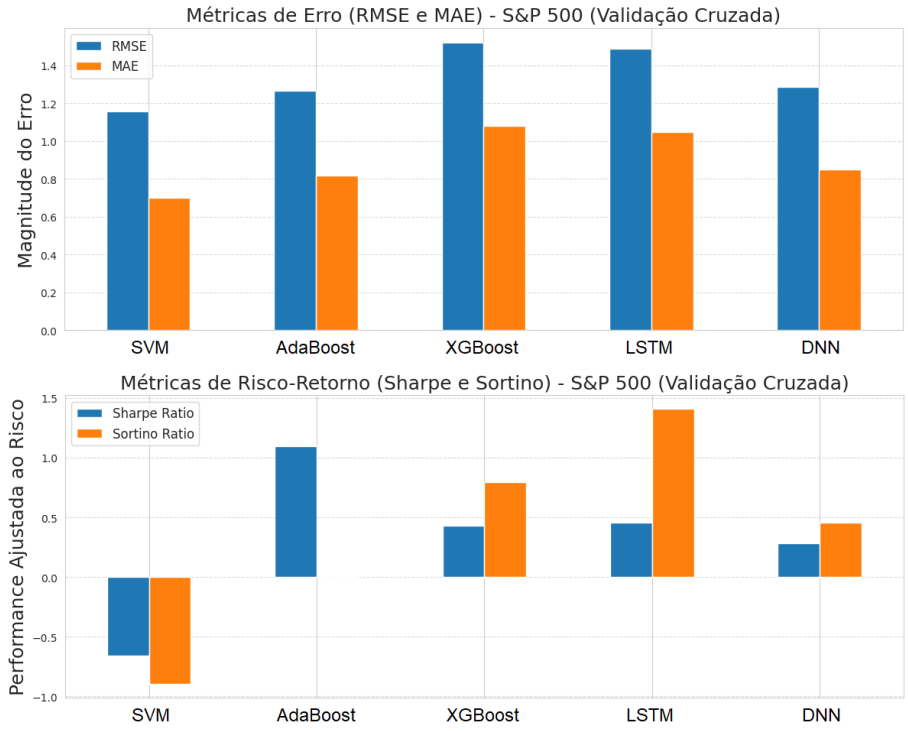

Figura 1. Desempenho dos algoritmos no índice S&P 500 com base em RMSE, MAE, *Sharpe* e *Sortino Ratio*.

Observa-se que o SVM obteve os menores valores de erro, demonstrando alta acurácia pontual. Contudo, seus *Sharpe* e *Sortino Ratios* foram negativos, sugerindo que, apesar da boa previsão dos retornos, a estratégia gerada não resultou em performance superior ao mercado. Em contrapartida, o *AdaBoost* apresentou erros mais elevados, porém com excelente desempenho ajustado ao risco, com *Sharpe Ratio* superior a 1.0. O *XGBoost* e o LSTM também se destacaram, apresentando combinações sólidas de acurácia e retorno ajustado ao risco.

Desempenho no Índice Nikkei 225

Na Figura 2, são apresentados os resultados dos algoritmos aplicados ao índice Nikkei 225, com foco nas mesmas métricas.

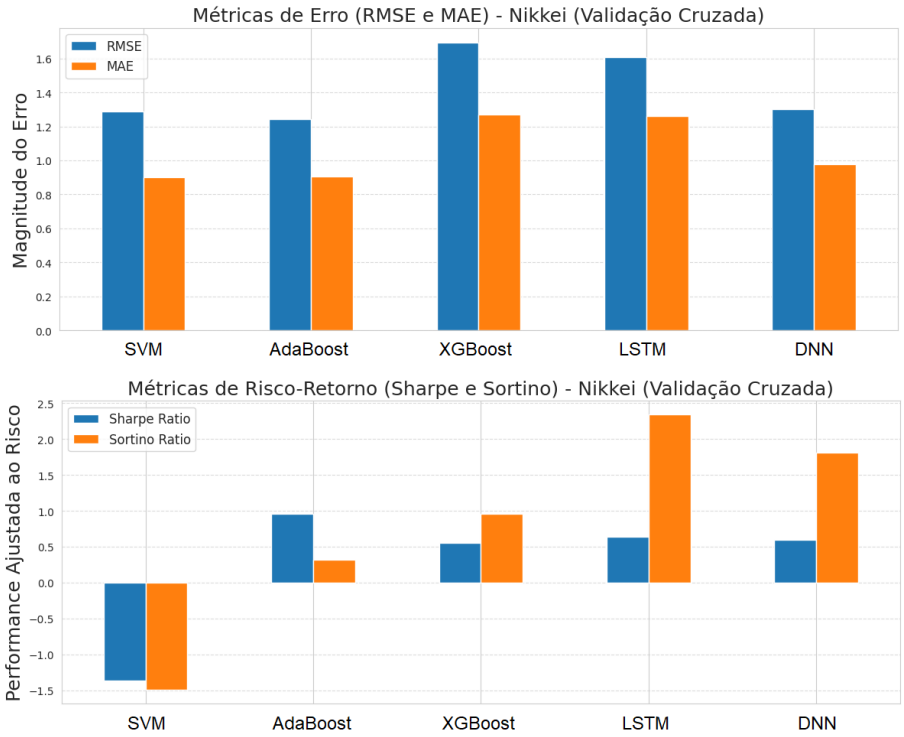

Figura 2. Desempenho dos algoritmos no índice Nikkei 225 com base em RMSE, MAE, *Sharpe* e *Sortino Ratio*.

Neste mercado, o LSTM obteve destaque com um *Sortino Ratio* superior a 2.5, revelando excelente capacidade de mitigar perdas em cenários voláteis. O *AdaBoost* manteve bom equilíbrio entre erro e risco-retorno. O SVM, apesar da boa acurácia, apresentou novamente métricas negativas de risco-retorno, enquanto o DNN teve desempenho neutro em todas as métricas. O *XGBoost* mostrou-se consistente, com valores positivos para *Sharpe* e *Sortino*, ainda que inferiores aos do LSTM.

Desempenho no Índice NASDAQ

A Figura 3 exibe os resultados obtidos para o índice NASDAQ, mantendo o mesmo conjunto de métricas.

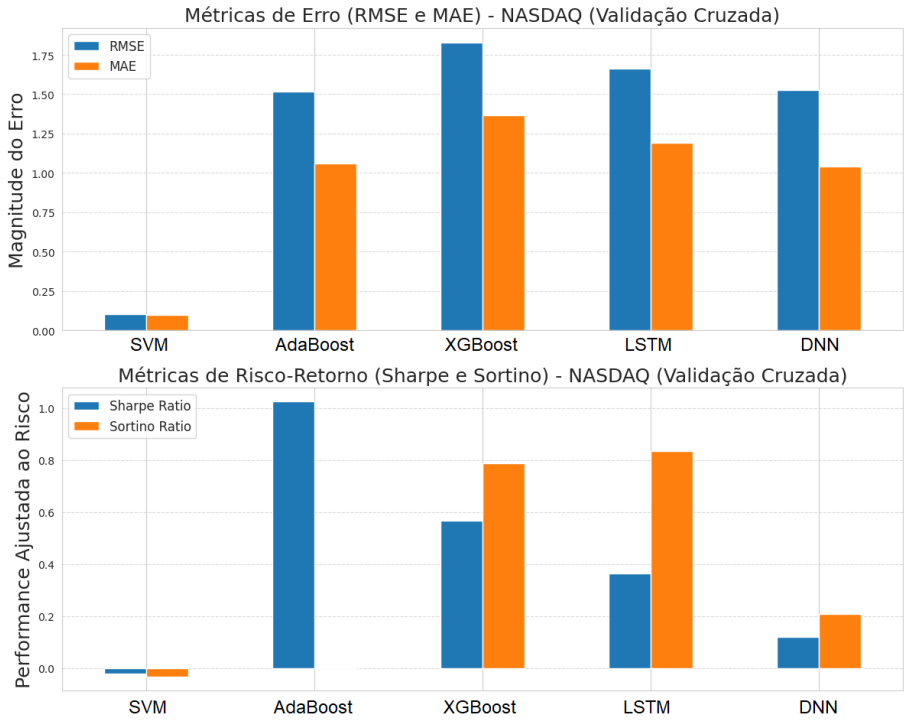

Figura 3. Desempenho dos algoritmos no índice NASDAQ com base em RMSE, MAE, Sharpe e Sortino Ratio.

No NASDAQ, os erros absolutos foram, em geral, mais elevados. O LSTM e o *AdaBoost* novamente se destacaram com métricas positivas de risco-retorno, apesar de apresentarem erros médios superiores aos do SVM. Este último manteve seu padrão de baixa efetividade em retorno ajustado ao risco, o que reforça a limitação de sua estratégia preditiva nesse mercado. O DNN apresentou desempenho intermediário, sem se destacar em nenhuma das métricas.

Desempenho da Programação Genética

A GP apresentou, de forma geral, os melhores resultados em termos de RMSE e MAE nas três bases analisadas, superando a maioria dos algoritmos comparados. Conforme mostra a Tabela 6, os menores valores de erro ocorreram no índice S&P 500, com RMSE de 0,7166 e MAE de 0,4800. No índice Nikkei, os valores foram mais elevados, embora ainda competitivos.

Tabela 6. Métricas de desempenho da Programação Genética.

Métrica	NASDAQ	S&P 500	Nikkei
RMSE	0.8901	0.7166	1.2553
MAE	0.5042	0.4800	0.6542
Sharpe Ratio	-0.0907	-0.0698	0.0157
Sortino Ratio	-0.0979	-0.0496	0.0147

No entanto, seus índices financeiros de desempenho ajustado ao risco ficaram negativos nos índices NASDAQ e S&P 500, e apenas levemente positivos no Nikkei. Isso indica que, apesar da boa precisão preditiva, a GP teve desempenho inferior na geração de retornos financeiros ajustados ao risco. Essa tendência pode ser visualizada de forma consolidada na Figura 4, que destaca o contraste entre a acurácia pontual e o retorno ajustado ao risco.

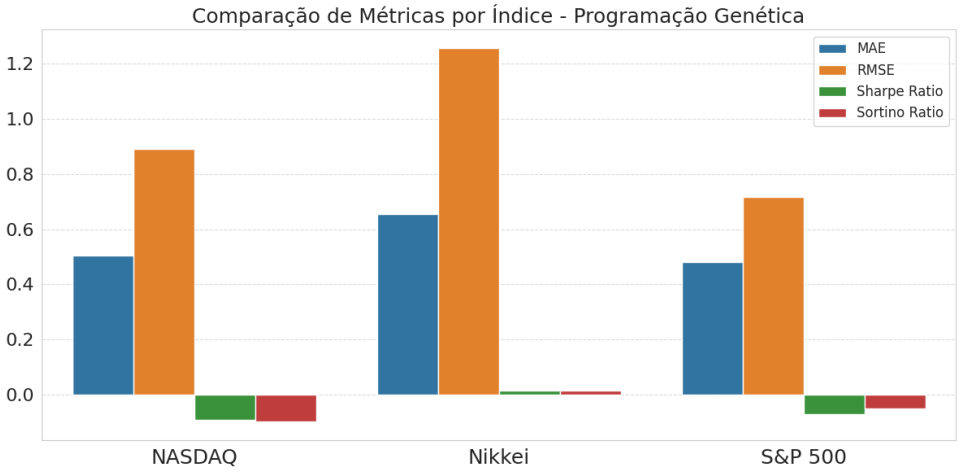

Figura 4. Desempenho da Programação Genética com base em RMSE, MAE, *Sharpe* e *Sortino Ratio*.

Síntese Comparativa

A análise dos três mercados indica que algoritmos como *AdaBoost* e LSTM, ainda que não apresentem a menor magnitude de erro, são superiores quando o objetivo envolve retorno ajustado ao risco. A GP, por sua vez, destacou-se por apresentar geralmente os melhores resultados em termos de RMSE e MAE, demonstrando alta precisão preditiva. Contudo, seus índices *Sharpe* e *Sortino* sugerem que o método de *trading* baseado na GP pode ser aprimorado para melhor aproveitamento do retorno ajustado ao risco.

Esses resultados demonstram a importância de se avaliar os modelos não apenas com base em acurácia pontual, mas considerando o risco envolvido nas decisões geradas. A escolha do modelo ideal depende diretamente do perfil de risco do investidor e do comportamento específico de cada mercado financeiro.

5. Conclusão

Este trabalho apresentou uma análise comparativa entre a GP e cinco algoritmos de ML na tarefa de previsão de tendências financeiras, utilizando os índices S&P 500, Nikkei 225 e NASDAQ. Para além das métricas tradicionais de erro, como RMSE e MAE, foram adotadas medidas ajustadas ao risco, como o *Sharpe Ratio* e o *Sortino Ratio*, aproximando a avaliação dos modelos ao contexto real da tomada de decisão no mercado financeiro.

Os resultados mostraram que algoritmos com maior acurácia pontual, como o SVM e a GP, não necessariamente produziram estratégias de investimento superiores. Por outro lado, abordagens como *AdaBoost* e LSTM mostraram maior consistência em retorno ajustado ao risco, especialmente em mercados voláteis.

Este estudo apresenta, no entanto, algumas limitações. A análise foi restrita a três índices amplos de mercado e não considerou ativos individuais. Os algoritmos foram avaliados de forma isolada, sem investigar arquiteturas híbridas ou estratégias de *ensemble*, e não houve ajuste automatizado de hiperparâmetros, o que pode ter limitado o desempenho de alguns modelos. Além disso, os critérios de geração de sinais de negociação adotaram limiares fixos, desconsiderando estratégias mais refinadas de alocação.

Como trabalhos futuros, propõe-se explorar modelos híbridos que combinem GP com redes neurais profundas ou métodos de *ensemble*, como em arquiteturas do tipo GP-LSTM ou GP-XGBoost. Também se sugere aplicar os modelos a ativos individuais, como ações e criptomoedas, e empregar técnicas de AutoML para ajuste dinâmico de parâmetros ao longo do processo evolutivo. Adicionalmente, recomenda-se investigar estratégias para garantir a reprodutibilidade em redes neurais profundas, como a definição de seeds para inicialização de pesos e controle da aleatoriedade em operações de treinamento, de modo a reduzir a variabilidade dos resultados entre execuções.

Referências

Atsalakis, G. S., Valavanis, K. P., and Emmanouilides, C. J. (2019). Surveying stock market forecasting techniques – part ii: Soft computing methods. *Expert Systems with Applications*, 36(3):5932–5941. Disponível na ScienceDirect.

Britannica Money (2025). Financial benchmarks — definition, examples, how to use. `https://www.britannica.com/money/financial-benchmarks`. Acessado em 21 de junho de 2025.

Chen, W. et al. (2021). Application of genetic programming in financial trading strategies: A comprehensive study. *Journal of Computational Finance*, 25(3):121–145.

Fischer, T. and Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669. Disponível na Elsevier.

Investopedia (2024). The difference between the sharpe ratio and the sortino ratio. Acessado em 2 de setembro de 2025.

Krauss, C., Do, X. A., and Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s&p 500. *The Journal of Financial Data Science*, 39(4):116–127. Disponível na SSRN.

Long, X., Kampouridis, M., and Jarchi, D. (2020). An in-depth investigation of genetic programming and nine other machine learning algorithms in a financial forecasting problem. *Applied Soft Computing*, 90:106188.

Mostafavi, S. (2025). Technical indicators and their predictive power for financial time series: An empirical study on s&p 500. *Journal of Financial Data Science*, 7(2):45–62.

Shen, J. et al. (2020). Hybrid models combining genetic programming and machine learning for financial forecasting. *Journal of Applied Soft Computing*, 92:106280.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Botstein, D., Altman, R. B., and Tibshirani, R. (2001). Missing value estimation methods for DNA microarray gene expression data. *Bioinformatics*, 17(6):520–525.