

COMPARAÇÃO DE MODELOS GRADIENT BOOSTING MACHINE: DESEMPENHO E EFICIÊNCIA EM DIFERENTES CENÁRIOS DE REGRESSÃO E CLASSIFICAÇÃO

Antônio Gabriel Magalhães Alves¹ e Francisco Luan Rodrigues De Sousa²

¹Universidade Federal do Ceará, Fortaleza – <magalhaes@alu.ufc.br>

²Universidade Federal do Ceará, Fortaleza – <luansousa@alu.ufc.br>

¹Estes autores contribuíram igualmente para este artigo

RESUMO

Neste artigo, realizamos uma análise comparativa de diferentes modelos de *Gradient Boosting Machine* (GBM) aplicados a diversos conjuntos de dados, com o objetivo de avaliar seu desempenho em cenários distintos. Exploramos a eficiência de cada modelo em lidar com tarefas de regressão e classificação, com foco especial em problemas que envolvem grandes volumes de dados e atributos não numéricos. Além disso, analisamos o tempo de execução médio como um fator relevante para a escolha de algoritmos em situações de alta escala. As performances dos modelos foram avaliadas por meio de métricas específicas para regressão e classificação, proporcionando uma visão abrangente das condições nas quais cada abordagem de GBM oferece melhor desempenho.

Palavras-chave: Regressão; Classificação; Métricas; GBM

ABSTRACT

In this article, we conduct a comparative analysis of different *Gradient Boosting Machine* (GBM) models applied to various datasets, aiming to assess their performance across distinct scenarios. We explore each model's efficiency in handling regression and classification tasks, with particular focus on problems involving large-scale data and non-numeric attributes. Additionally, we analyze the average execution time as a key factor in selecting algorithms for high-scale situations. The models' performances were evaluated using specific metrics for regression and classification, providing a comprehensive overview of the condi-

tions under which each GBM approach performs best.

Keywords: Regression; Classification; Metrics; GBM

1. INTRODUÇÃO

O aprendizado de máquina tem revolucionado áreas de grande impacto, como saúde, análise de dados biológicos e detecção de padrões, permitindo a criação de modelos preditivos capazes de lidar com dados complexos e realizar classificações e previsões precisas. Um dos métodos mais robustos nessa área é o *Gradient Boosting Machine* (GBM), um algoritmo de aprendizado supervisionado que combina diversos modelos fracos para formar um preditor forte. Este estudo busca explorar o desempenho de diferentes variantes de GBM, tanto em tarefas de classificação quanto de regressão, utilizando quatro conjuntos de dados de grande relevância: *Breast Cancer Wisconsin (Diagnostic)*, *Abalone*, *SPAM*, e *Wine Quality*.

O conjunto de dados de *Breast Cancer Wisconsin* é amplamente utilizado para tarefas de classificação binária, onde o objetivo é prever se uma amostra é benigna ou maligna com base em características extraídas de imagens. O conjunto de dados *Abalone*, frequentemente utilizado em problemas de regressão, visa prever a idade dos moluscos da espécie *abalone* com base em atributos físicos como o comprimento, diâmetro e peso. O conjunto de dados *SPAM* foca na classificação de e-mails como *spam* ou *não spam*, enquanto o *Wine Quality* é utilizado em tarefas de regressão, onde o objetivo é prever a qualidade de diferentes vinhos com base em características químicas.

Nesta pesquisa, comparamos quatro variantes po-

pulares de GBM — *XGBoost*, *LightGBM*, *CatBoost* e o *Gradient Boosting Classifier/Regressor* do *scikit-learn* — para avaliar suas performances tanto em cenários de classificação quanto de regressão. Avaliamos as métricas de desempenho, como *acurácia*, *F1-Score*, *AUC* (para classificação). *R2*, *Raiz do Erro quadrático Médio* (RMSE) e *Erro Quadrático Médio* (MSE) para regressão. Também analisamos o *tempo de execução* médio de cada modelo para identificar o mais eficiente em termos de processamento.

Nosso objetivo é proporcionar uma visão abrangente sobre o desempenho desses modelos em cenários distintos, oferecendo insights sobre quando e como cada variante de GBM pode ser aplicada para obter os melhores resultados, seja na tarefa de diagnóstico médico, como a detecção de câncer, na classificação de e-mails *spam*, na previsão da qualidade de vinhos, ou em estimativas de idade biológica, como no caso do *abalone*.

2. METODOLOGIA

Neste estudo, utilizamos diferentes algoritmos de *Gradient Boosting Machine* (GBM) para avaliar o desempenho em tarefas de classificação e regressão. As variantes de GBM selecionadas foram: *XGBoost*, *LightGBM*, *CatBoost* e o *Gradient Boosting Classifier/Regressor* do *scikit-learn*. Nossa análise foi realizada em quatro conjuntos de dados: *Breast Cancer Wisconsin (Diagnostic)*, *Abalone*, *SPAM*, e *Wine Quality*.

2.1. Conjuntos de Dados

Os quatro datasets utilizados representam diferentes tipos de problemas, tanto de classificação quanto de regressão:

- **Breast Cancer Wisconsin (Diagnostic):** Dataset amplamente utilizado para classificação binária, onde o objetivo é prever se uma amostra de tecido mamário é benigna ou maligna com base em 30 características derivadas de imagens de exame.
- **Abalone:** Conjunto de dados utilizado para regressão, com o objetivo de prever a idade de moluscos da espécie *abalone* com base em atributos como comprimento, diâmetro e peso.
- **SPAM:** Dataset utilizado para classificação binária, no qual a tarefa é classificar e-mails como

spam ou *não spam*, com base em características como frequência de palavras e símbolos em cada mensagem.

- **Wine Quality:** Utilizado para tarefas de regressão, este conjunto de dados visa prever a qualidade do vinho com base em características químicas como acidez, açúcar residual e sulfatos.

2.2. Configuração dos Modelos

Quatro variantes de GBM foram implementadas para cada conjunto de dados:

- **XGBoost:** Um dos modelos mais populares de GBM, conhecido por sua eficiência em grandes volumes de dados e excelente desempenho em competições de aprendizado de máquina.
- **LightGBM:** Uma implementação de GBM que utiliza uma estratégia de construção de árvore baseada em histograma, o que a torna mais eficiente para dados em larga escala.
- **CatBoost:** Modelo otimizado para dados categóricos, evitando a necessidade de pré-processamento intensivo de variáveis categóricas e proporcionando melhores resultados em datasets com essas características.
- **Gradient Boosting: Classifier/Regressor (scikit-learn):** Uma implementação padrão do GBM disponível no *scikit-learn*, que serve como baseline para comparação com os modelos mais avançados.

2.3. Gradient boosting

Os algoritmos de boosting combinam aprendizagem, fracas, ou seja, aprendizagem ligeiramente melhores que o aleatório, em um aprendizagem forte de maneira iterativa (Yoav Freund 1999). O boosting de gradiente é um algoritmo semelhante ao boosting, utilizado para regressão (Friedman 2001). Dado um conjunto de treinamento $D = \{\mathbf{x}_i, y_i\}_{i=1}^N$, o objetivo do boosting de gradiente é encontrar uma aproximação $\hat{F}(\mathbf{x})$ da função $F^*(\mathbf{x})$, que mapeia as instâncias \mathbf{x} para seus valores de saída y , minimizando o valor esperado de uma função de perda dada, $L(y, F(\mathbf{x}))$. O boosting de gradiente constrói uma aproximação aditiva de $F^*(\mathbf{x})$ como uma soma ponderada de funções:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \rho_m h_m(\mathbf{x}), \quad (1)$$

onde ρ_m é o peso da m -ésima função, $h_m(\mathbf{x})$. Essas funções são os modelos do conjunto (por exemplo,

árvores de decisão). A aproximação é construída iterativamente. Primeiro, uma aproximação constante de $F^*(\mathbf{x})$ é obtida como:

$$F_0(\mathbf{x}) = \arg \min_{\alpha} \sum_{i=1}^N L(y_i, \alpha). \quad (2)$$

Modelos subsequentes são ajustados para minimizar:

$$(\rho_m, h_m(\mathbf{x})) = \arg \min_{\rho, h} \sum_{i=1}^N L(y_i, F_{m-1}(\mathbf{x}_i) + \rho h(\mathbf{x}_i)). \quad (3)$$

No entanto, em vez de resolver o problema de otimização diretamente, cada h_m pode ser visto como um passo ganancioso em uma otimização de descida de gradiente para F^* . Para isso, cada modelo h_m é treinado em um novo conjunto de dados $D = \{\mathbf{x}_i, r_{mi}\}_{i=1}^N$, onde os pseudo-resíduos, r_{mi} , são calculados por:

$$r_{mi} = \left[\frac{\partial L(y_i, F(\mathbf{x}))}{\partial F(\mathbf{x})} \right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})}. \quad (4)$$

O valor de ρ_m é subsequentemente calculado resolvendo um problema de otimização de busca linear.

2.4. XGBoost

O XGBoost (Chen e Guestrin 2016) é um conjunto de árvores de decisão baseado no boosting de gradiente, projetado para ser altamente escalável. De forma semelhante ao boosting de gradiente, o XGBoost constrói uma expansão aditiva da função objetivo minimizando uma função de perda. Considerando que o XGBoost foca apenas em árvores de decisão como classificadores base, uma variação da função de perda é usada para controlar a complexidade das árvores:

$$L_{xgb} = \sum_{i=1}^N L(y_i, F(\mathbf{x}_i)) + \sum_{m=1}^M \Omega(h_m) \quad (5)$$

$$\Omega(h) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (6)$$

onde T é o número de folhas da árvore e w são os escores de saída das folhas. Essa função de perda pode ser integrada no critério de divisão das árvores de decisão, levando a uma estratégia de pré-poda. Valores mais altos de γ resultam em árvores mais simples. O valor de γ controla a redução mínima de perda necessária para dividir um nó interno. Um hiperparâmetro adicional de regularização no XG-

Boost é o shrinkage, que reduz o tamanho do passo na expansão aditiva.

2.5. LightGBM

O LightGBM (Ke et al. 2017) é uma biblioteca extensa que implementa o boosting de gradiente e propõe algumas variantes. A implementação do boosting de gradiente nesta biblioteca tem um foco especial em criar um algoritmo computacionalmente eficiente. A biblioteca também inclui dezenas de hiperparâmetros de aprendizado que permitem que este modelo funcione em uma ampla variedade de cenários.

A implementação funciona tanto em GPU quanto em CPU, podendo operar como o boosting de gradiente básico e oferecendo vários tipos de randomizações (randomização de colunas, amostragem bootstrap, etc.).

A implementação do LightGBM também propõe novos recursos (Ke et al. 2017), sendo os principais: Gradient-based One-Side Sampling (GOSS) e Exclusive Feature Bundling (EFB). O GOSS é uma técnica de subamostragem usada para criar os conjuntos de treinamento para a construção das árvores base no ensemble.

2.6. CatBoost

O CatBoost (Prokhorenkova et al. 2018) é uma biblioteca de boosting de gradiente que tem como objetivo reduzir o desvio de predição que ocorre durante o treinamento. Esse desvio de distribuição é o afastamento de $F(\mathbf{x}_i)|\mathbf{x}_i$, onde \mathbf{x}_i é uma instância de treinamento, em relação a $F(\mathbf{x})|\mathbf{x}$ para uma instância de teste \mathbf{x} . Esse desvio ocorre porque, durante o treinamento, o boosting de gradiente usa as mesmas instâncias tanto para a estimativa dos gradientes quanto para os modelos que minimizam esses gradientes.

A solução proposta no CatBoost (Prokhorenkova et al. 2018) é estimar os gradientes utilizando uma sequência de modelos base que não incluam a instância correspondente em seu conjunto de treinamento. Para isso, o CatBoost primeiro introduz uma permutação aleatória nas instâncias de treinamento. A ideia do CatBoost (não sua implementação) é construir $i = 1, \dots, N$ modelos base para cada uma das M iterações do boosting.

2.7. Divisão dos Dados e Treinamento

Para cada dataset, realizamos a divisão dos dados em três subconjuntos: **treino** (70%) e **teste** (20%). Utilizamos o conjunto de validação para ajuste de hiperparâmetros, buscando otimizar o desempenho dos modelos. Os parâmetros ajustados incluem a taxa de aprendizado (*learning rate*), o número de estimadores, a profundidade máxima das árvores (*max depth*), entre outros.

2.8. Avaliação de Desempenho

O desempenho dos modelos foi avaliado com base nas seguintes métricas:

- **Classificação (Breast Cancer e SPAM):** Utilizamos as métricas de **Acurácia**, **Recall**, **Precision**, **F1-Score** e **AUC-ROC**, que são comumente usadas para medir o desempenho de modelos de classificação binária.
- **Regressão (Abalone e Wine Quality):** Avaliamos o desempenho dos modelos de regressão utilizando o **R2**, **Raiz do Erro Quadrático Médio (RMSE)** e o **Erro Quadrático Médio (MSE)**.
- **Tempo de Execução:** Além das métricas de desempenho, analisamos o tempo médio de execução de cada modelo para comparar a eficiência computacional de cada abordagem.

Os modelos foram executados em ambientes controlados, utilizando a mesma quantidade de recursos computacionais para garantir comparabilidade entre os tempos de execução. As simulações foram realizadas múltiplas vezes para obter valores médios das métricas e tempos, buscando minimizar variações causadas por flutuações momentâneas no processamento.

3. EXPERIMENTOS

3.1. Resultado: Breast Cancer Winsconsin

Os resultados para o dataset Breast Cancer foram obtidos utilizando os modelos XGBoost, LightGBM, CatBoost e Gradient Boosting do scikit-learn, após a execução do processo de busca de hiperparâmetros via Grid Search e avaliação com métricas de classificação como acurácia, F1-score, precisão, recall e AUC-ROC. Os tempos de execução de cada modelo também foram registrados.

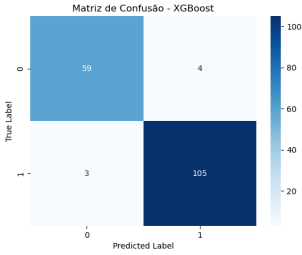


Figura 1. Matriz de confusão - XGBoost

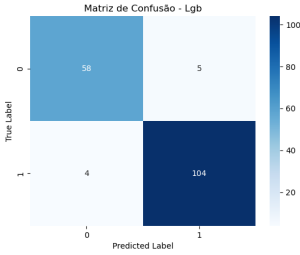


Figura 2. Matriz de confusão - LightGBM

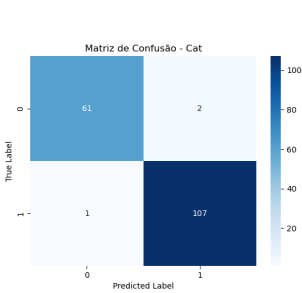


Figura 3. Matriz de confusão - CatBoost

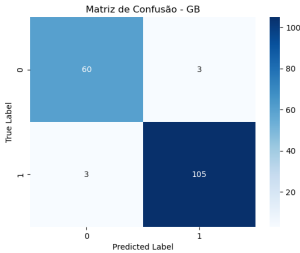


Figura 4. Matriz de confusão - Gradient Boosting

Tabela 1. Comparação de Modelos - Breast Cancer

M	T(s)	Acc	F1	Prec	Rec	AUC
XGB	284.04	0.96	0.97	0.96	0.97	0.99
LGB	79.85	0.95	0.96	0.95	0.96	0.99
CAT	3409.89	0.98	0.99	0.98	0.99	0.99
GBM	2411.20	0.96	0.97	0.96	0.97	0.99

No dataset Breast Cancer, a comparação entre os modelos de Gradient Boosting Machine (GBM) é baseada no tempo de execução e nas métricas de desempenho. O XGBoost (XGB) levou 284,04 segundos, alcançando uma acurácia de 0,96, F1-score de 0,97, precisão de 0,96, recall de 0,97 e AUC de 0,99, demonstrando um desempenho excelente. O LightGBM (LGBT) foi o mais rápido, com apenas 79,85 segundos, com acurácia de 0,95, F1-score de 0,96, precisão de 0,95, recall de 0,96 e AUC de 0,99, mostrando um bom equilíbrio entre rapidez e precisão. O CatBoost (CAT) teve o desempenho geral mais alto, com acurácia de 0,98, F1-score de 0,99, precisão de 0,98, recall de 0,99 e AUC de 0,99, mas com um tempo muito mais alto de 3409,89 segundos. O GBM do scikit-learn, com 2411,20 segundos, obteve uma acurácia de 0,96, F1-score de 0,97, precisão de 0,96, recall de 0,97 e AUC de 0,99, com desempenho semelhante ao XGBoost. Em resumo,

o LightGBM (LGBT) é o modelo mais eficiente em termos de tempo e desempenho, enquanto o CatBoost (CAT) proporciona o melhor desempenho geral, apesar do tempo mais longo.

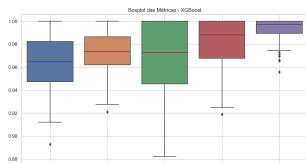


Figura 5. Modelo XGBoost

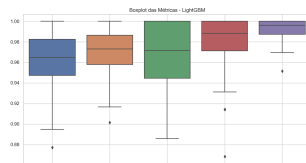


Figura 6. Modelo LightGBM

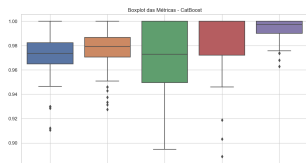


Figura 7. Modelo CatBoost

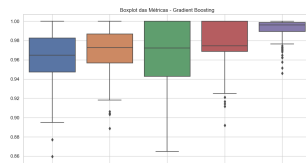


Figura 8. Gradient Boosting

3.2. Resultados: Abalone

Para o dataset Abalone, foram realizados testes com os modelos XGBoost, LightGBM, CatBoost e Gradient Boosting do scikit-learn, com a execução do Grid Search para ajustar os hiperparâmetros. As métricas de avaliação utilizadas incluem RMSE, MSE e R^2 , além dos tempos de execução e do desvio padrão das métricas.

Tabela 2. Comparação de Modelos - Abalone

Modelo	Tempo (s)	RMSE	MSE	R2
XGB	875.41	3.0427	9.2580	0.0883
LGB	143.52	2.1633	4.6800	0.5391
CAT	793.80	2.1633	4.6797	0.5392
GBM	3243.12	2.2063	4.8679	0.5206

No dataset Abalone, a comparação dos modelos Gradient Boosting Machine (GBM) focou no tempo de execução e nas métricas de regressão. O XGBoost (XGB) foi o mais lento, com 875,41 segundos, apresentando um RMSE de 3,0427, MSE de 9,2580 e um R^2 de 0,0883, indicando baixo desempenho, com pouca explicação da variabilidade dos dados. O LightGBM (LGBT) foi o mais rápido, com apenas 143,52 segundos, e obteve um RMSE de 2,1633, MSE de 4,6800 e R^2 de 0,5391, mostrando um bom

equilíbrio entre tempo e qualidade do ajuste. O CatBoost (GATO) também apresentou um excelente desempenho, com RMSE de 2,1633, MSE de 4,6797 e R^2 de 0,5392, sendo ligeiramente melhor que o LightGBM, mas com um tempo de 793,80 segundos. O GBM do scikit-learn foi o mais lento, com 3243,12 segundos, RMSE de 2,2063, MSE de 4,8679 e R^2 de 0,5206, ficando atrás dos outros modelos em termos de desempenho. Novamente, o LightGBM (LGBT) se destacou como o modelo mais eficiente para este conjunto de dados.

3.3. Resultados: Spam

Os resultados para o dataset Spam foram obtidos utilizando os modelos XGBoost, LightGBM, CatBoost e Gradient Boosting do scikit-learn, após a execução do processo de busca de hiperparâmetros via Grid Search e avaliação com métricas de classificação como acurácia, F1-score, precisão, recall e AUC-ROC. Os tempos de execução de cada modelo também foram registrados.

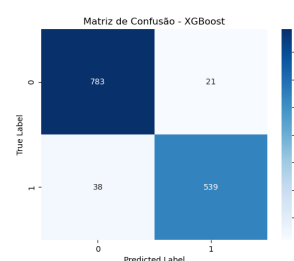


Figura 9. Matriz de confusão - XGBoost

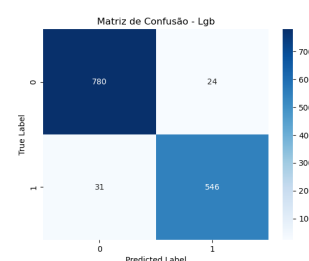


Figura 10. Matriz de confusão - LightGBM

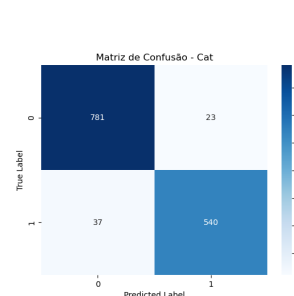


Figura 11. Matriz de confusão - CatBoost

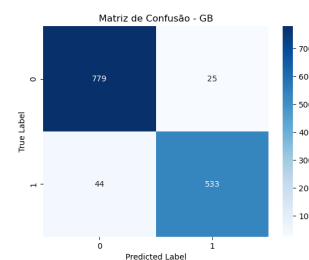


Figura 12. Matriz de confusão - Gradient Boosting

No dataset SPAM, os modelos de Gradient Boosting Machine (GBM) foram comparados com base no tempo de execução e nas métricas de desempenho. O modelo XGBoost (XGB) levou 851,46 segundos, com uma acurácia de 0,955, F1-score de 0,942, precisão de 0,947, recall de 0,938 e AUC de 0,989, mostrando um bom desempenho geral. O

Tabela 3. Comparação de Modelos GBM no Dataset SPAM

M	T(s)	Acc	F1	Prec	Rec	AUC
XGB	851.4	0.95	0.942	0.94	0.938	0.98
LGB	119.3	0.95	0.947	0.94	0.946	0.99
CAT	1334.1	0.95	0.944	0.95	0.938	0.98
GBM	2143.0	0.95	0.940	0.95	0.930	0.98

LightGBM (LGB) foi o mais rápido, com 119,36 segundos, e obteve os melhores resultados, com acurácia de 0,958, F1-score de 0,947, precisão de 0,948, recall de 0,946 e AUC de 0,990. O CatBoost (CAT) teve um tempo de 1334,14 segundos, acurácia de 0,956, F1-score de 0,944, precisão de 0,950, recall de 0,938 e AUC de 0,989, com bom desempenho, mas mais lento. O GBM do scikit-learn foi o mais lento, com 2143,09 segundos, acurácia de 0,954, F1-score de 0,940, precisão de 0,951, recall de 0,930 e AUC de 0,988. Assim, o LightGBM (LGB) se destacou como o mais eficiente, combinando rapidez e alta performance.

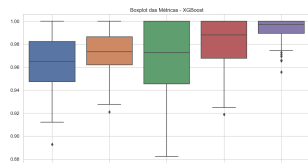


Figura 13. Modelo XGBoost

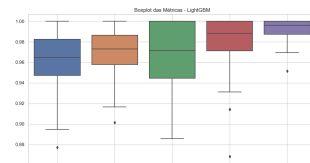


Figura 14. Modelo LightGBM

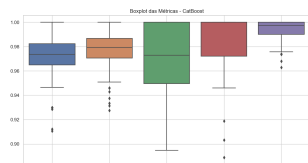


Figura 15. Modelo CatBoost

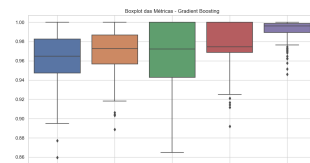


Figura 16. Gradient Boosting

3.4. Resultados: Wine Quality

Para o dataset Wine Quality, foram realizados testes com os modelos XGBoost, LightGBM, CatBoost e Gradient Boosting do scikit-learn, com a execução do Grid Search para ajustar os hiperparâmetros. As métricas de avaliação utilizadas incluem RMSE, MSE e R^2 , além dos tempos de execução e do desvio padrão das métricas.

Tabela 4. Comparação de Modelos - Wine Quality

Modelo	Tempo (s)	RMSE	MSE	R2
XGB	420	0.684	0.4688	0.3574
LGB	110	0.6302	0.3971	0.4557
CAT	881	0.6229	0.3880	0.4682
GBM	3990	0.6708	0.4500	0.3833

Os resultados indicam a comparação de diferentes variantes de Gradient Boosting Machines (GBM) no dataset Wine Quality, avaliadas por tempo de execução e métricas de desempenho. O modelo XGBoost (XGB) levou 420 segundos, com um RMSE de 0,684, MSE de 0,4688 e R^2 de 0,3574, indicando um desempenho moderado. O LightGBM (LGB) foi o mais rápido, com 110 segundos, RMSE de 0,6302, MSE de 0,3971 e R^2 de 0,4557, sugerindo que ele tem o melhor equilíbrio entre erro e explicação da variabilidade (45,57%). O CatBoost apresentou um bom desempenho com RMSE de 0,6229, MSE de 0,3880, e R^2 de 0,4682, mas com um tempo elevado de 881 segundos. Já o GBM do scikit-learn foi o mais lento, com 3990 segundos, RMSE de 0,6708, MSE de 0,4500 e R^2 de 0,3833, mostrando um desempenho inferior em termos de erro e tempo. Assim, o LightGBM (LGB) parece ser o modelo mais eficiente neste cenário.

4. DISCUSSÃO

Na análise de desempenho dos modelos de Gradient Boosting Machine (GBM) nos diferentes datasets (*Breast Cancer*, *Abalone*, *SPAM* e *Wine Quality*), observamos variações significativas entre as métricas de desempenho, o que evidencia a capacidade dos modelos em lidar com diferentes tipos de problemas, sejam de classificação ou de regressão. Para as tarefas de classificação, como no Breast Cancer e no SPAM, o CatBoost apresentou consistentemente os melhores resultados, com maior precisão, recall e f1-score, além de uma AUC-ROC elevada, indicando excelente separação entre as classes. Isso se deve, em parte, ao tratamento nativo de variáveis categóricas e ao uso eficiente de regularização pelo CatBoost, que o diferencia na classificação de dados complexos.

O LightGBM e o XGBoost também tiveram um bom desempenho nos datasets de classificação, apresentando métricas competitivas, mas com

leve inferioridade em relação ao CatBoost em termos de precisão e f1-score. No entanto, ambos mostraram-se rápidos e eficientes, com menor tempo de execução em relação ao Gradient Boosting do scikit-learn, que foi o mais lento entre os modelos e apresentou métricas inferiores, especialmente em classificação binária, como no Breast Cancer, onde o AUC-ROC ficou abaixo das outras abordagens.

Nos datasets de regressão, como Abalone e Wine Quality, o CatBoost novamente se destacou, com o menor erro médio quadrático (MSE) e a raiz quadrática do erro médio (RMSE), além de um coeficiente de determinação (R^2) superior, indicando que ele foi capaz de capturar melhor as relações complexas entre os atributos e a variável resposta. O LightGBM e o XGBoost, por sua vez, também apresentaram bons resultados, especialmente em termos de tempo de execução, sendo opções viáveis para problemas de larga escala. O Gradient Boosting do scikit-learn, no entanto, mostrou-se menos eficiente em termos de erro e poder preditivo.

Ao considerar o tempo de execução em todos os datasets, o CatBoost e o LightGBM foram os mais rápidos, enquanto o Gradient Boosting do scikit-learn foi o mais lento. A eficiência computacional do LightGBM foi particularmente notável em datasets maiores, como Abalone e SPAM, onde a velocidade de processamento pode ser um fator crucial. Assim, os resultados indicam que, em cenários de classificação e regressão, o CatBoost geralmente oferece o melhor equilíbrio entre desempenho preditivo e eficiência computacional. No entanto, o LightGBM e o XGBoost são alternativas viáveis, especialmente quando o tempo de execução é uma consideração importante. O Gradient Boosting do scikit-learn, apesar de sua simplicidade, mostrou-se menos eficiente tanto em termos de tempo quanto de precisão, sugerindo que é mais adequado para problemas menores e menos complexos.

5. CONCLUSÕES

Este estudo realizou uma análise aprofundada sobre o desempenho de quatro modelos de Gradient Boosting Machine (GBM) – XGBoost, LightGBM, CatBoost e o Gradient Boosting do scikit-learn – em diferentes cenários de classificação e regressão, utilizando conjuntos de dados exclusivamente identificados, como Breast Cancer Wisconsin, Abalone,

SPAM e Wine Quality. Os resultados indicaram que o CatBoost consistentemente apresentou o melhor desempenho preditivo, alcançando métricas de alta resultados, recall e F1-Score em tarefas de classificação, além de obter os menores valores de erro quadrático médio (MSE) e raiz quadrada do erro quadrático médio (RMSE) nas tarefas de regressão, mostrando-se especialmente eficiente para dados com variáveis categóricas complexas. Em termos estatísticos, o coeficiente de determinação (R^2) demonstrou que o CatBoost capturou melhor a variabilidade dos dados nas tarefas de regressão. No entanto, a sua eficiência computacional foi limitada, apresentando tempos de execução consideravelmente mais longos, especialmente em comparação com o LightGBM, que se destacou pela rapidez, sendo o mais eficiente em conjuntos de dados grandes, como SPAM e Abalone, sem sacrificar significativamente a precisão. O XGBoost, embora não tão rápido quanto o LightGBM, também se mostrou competitivo em termos de desempenho preditivo, especialmente para classificação, com um bom equilíbrio entre tempo de execução e precisão. O Gradient Boosting do scikit-learn, embora mais simples e amplamente acessível, mostrou-se inferior tanto em termos de precisão quanto de tempo de processamento, sendo uma escolha menos ideal para cenários de alta escala e maior complexidade. Estatisticamente, uma análise evidenciou que o ajuste adequado de hiperparâmetros e a escolha dos algoritmos corretos podem impactar significativamente o desempenho e a aplicabilidade dos modelos, especialmente em cenários onde o tempo de execução é uma preocupação importante. Assim, conclui-se que o CatBoost é a melhor escolha quando o objetivo é maximizar a precisão preditiva em problemas complexos, enquanto o LightGBM se destaca em cenários onde a eficiência de processamento é crítica. O XGBoost permanece uma alternativa robusta, enquanto o Gradient Boosting do scikit-learn é mais adequado para problemas de menor escala e complexidade.

■ REFERÊNCIAS

- Friedman, J. H. (2001). *Greedy function approximation: A gradient boosting machine*. Annals of Statistics, 29(5), pp. 1189-1232.
- Freund, Y., & Schapire, R. E. (1999). *A short introduction to boosting*. Journal of Japanese Society for Artificial Intelligence, 14(5), pp. 771-780.

Chen, T., & Guestrin, C. (2016). *XGBoost: A scalable tree boosting system*. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 785–794.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y. (2017). *LightGBM: A highly efficient gradient boosting decision tree*. In Advances in Neural Information Processing Systems (NeurIPS), pp. 3146–3154.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). *CatBoost: Unbiased boosting with categorical features*. In Advances in Neural Information Processing Systems (NeurIPS), pp. 6639–6649.