

Explorando Modelos Preditivos de Tempo de Execução de Simulações de Reservatório em Clusters HPC*

Alan L. Nunes¹✉, Lucas A. F. da Costa³, Felipe A. Portella²✉,
José Viterbo¹, Leonardo V. Neri³, Paulo J. B. Estrela², Luciana de C. Franci³,
Renzo Q. Malini², Lúcia Maria de Assumpção Drummond¹

`alan.lira@id.uff.br {viterbo, lucia}@ic.uff.br
{felipeportella, paulo.estrela, renzo}@petrobras.com.br
lvn@cesar.school {lafc, lcf2}@cesar.org.br`

¹ Instituto de Computação – Universidade Federal Fluminense (UFF), Niterói – Brasil

²Petróleo Brasileiro S.A. (PETROBRAS), Rio de Janeiro – Brasil

³Centro de Estudos e Sistemas Avançados do Recife (CESAR), Recife – Brasil

Abstract. This work evaluates machine learning models to predict the execution time of petroleum reservoir simulation jobs on HPC clusters. Using real-world logs from Petrobras, we assessed the XGBoost and J48 algorithms with different discretization schemes for the target attribute. Our results demonstrate that discretizations with broader classes offer greater robustness and a significantly lower and stable percentage error over time, while fine-grained classes generate higher accuracy and higher percentage errors.

Resumo. Este trabalho avalia modelos de aprendizado de máquina para a predição do tempo de execução de jobs de simulação de reservatórios petrolíferos em clusters HPC. Utilizando logs reais da Petrobras, testamos os algoritmos XGBoost e J48 com diferentes esquemas de discretização do atributo alvo. Nossos resultados demonstram que discretizações com classes mais amplas oferecem maior robustez e erro percentual significativamente menor e estável ao longo do tempo, enquanto classes granulares geram maior acurácia e erros percentuais mais elevados.

1. Introdução

A simulação de reservatórios é uma ferramenta essencial na indústria de óleo e gás, usada para reduzir riscos e otimizar decisões. Seu principal objetivo é reproduzir o histórico de produção e prever o desempenho futuro, fornecendo *insights* sobre o comportamento dos fluidos [Portella et al. 2022]. Seus modelos utilizam malhas 3D complexas para representar o reservatório, com formulações baseadas em princípios físicos como conservação de massa e a Lei de Darcy [Coats 1982]. Devido à sua alta complexidade computacional, as simulações são executadas em *clusters* de alto desempenho (HPC), sendo necessárias múltiplas execuções para quantificar as incertezas do reservatório e dos fluidos.

Algoritmos de aprendizado de máquina podem aumentar a eficiência de *clusters* HPC, permitindo que escalonadores refinem políticas de alocação a partir da previsão do

*Os autores agradecem à Petrobras pelo financiamento deste trabalho.

tempo de execução de *jobs* [Nunes et al. 2024]. No entanto, sua adoção costuma ser limitada por três fatores: escassez de dados representativos, mudança no perfil de *jobs* e a incerteza quanto a ganhos de eficiência através das previsões [Kuchnik et al. 2019].

O aprendizado de máquina supervisionado é frequentemente usado para prever o tempo de execução de *jobs*, onde um algoritmo é treinado com um conjunto de dados rotulados para fazer previsões sobre novos dados [Mohri et al. 2018]. O método de regressão prevê valores numéricos, como o tempo de execução, sendo avaliado por métricas como MAE (erro absoluto médio) e MAPE (erro percentual médio absoluto). Já a classificação categoriza o tempo de execução em intervalos predefinidos, sendo avaliada pela acurácia (*i.e.*, proporção de previsões corretas) [Nunes et al. 2023, Lima et al. 2024].

Neste trabalho avaliamos diferentes modelos de aprendizado de máquina para a previsão do tempo de execução de *jobs* de simulação de reservatórios petrolíferos.

2. Resultados Experimentais

Os experimentos foram conduzidos a partir de um conjunto de dados real, proveniente de *logs* do gerenciador de *jobs* Slurm do ambiente de HPC da Petrobras, abrangendo o período de junho de 2022 a junho de 2023. O conjunto de dados final, obtido após etapas de filtragem e pré-processamento ([Nunes et al. 2025]), consiste em 1.305.452 registros de *jobs*. Destes, 1.174.186 *jobs* (90%), executados entre junho de 2022 e maio de 2023, foram utilizados para treinamento e validação cruzada, enquanto 131.266 *jobs* (10%), executados em junho de 2023, foram reservados para teste, caracterizando uma avaliação *Out-of-Sample* que mimetiza um cenário de implantação realista, onde o modelo prediz eventos futuros com base em dados históricos, sem vazamento de informação.

Foram avaliados dois modelos de classificação, tendo como objetivo a previsão da classe de duração dos *jobs*: **XGBoost** [Chen and Guestrin 2016], um método de *gradient boosting* conhecido por seu alto desempenho, e **J48**, uma implementação da árvore de decisão C4.5 [Quinlan 2014]. Para lidar com atributos categóricos, duas estratégias de codificação foram comparadas: **Target Encoder**, que codifica categorias com a média de valores do atributo alvo, e **Ordinal Encoder**, que atribui um valor inteiro arbitrário a cada categoria. Adicionalmente, foram investigados dois esquemas de discretização do tempo de execução em classes do atributo alvo (*elapsed*), conforme descrito nas Tabelas 1 e 2.

2.1. Análise dos Resultados: Conjunto de Teste Completo

A Tabela 3 sumariza o desempenho preditivo dos modelos no conjunto de teste, reportando as métricas de **Acurácia**, **MAE** e **MAPE**. Para o cálculo do MAE e MAPE, o valor numérico foi definido como o limite inferior do intervalo de tempo da classe prevista.

Os resultados indicam um *trade-off* claro entre a acurácia e a magnitude do erro percentual, intimamente ligado à escolha do esquema de discretização. O modelo **XG-Boost + Target Encoder + Classes UFF** obteve o melhor desempenho em termos de acurácia (0,68) e MAE (≈ 8642), demonstrando a eficácia do algoritmo e da codificação em capturar padrões complexos nos dados quando classes mais granulares são utilizadas. Contudo, seu MAPE relativamente alto (131,58%) revela uma limitação na precisão relativa, particularmente para *jobs* de longa duração, onde erros absolutos similares resultam em maiores discrepâncias percentuais.

Tabela 1. Primeiro esquema de discretização do atributo alvo (Classes UFF).

Classe	Intervalo de Duração (em segundos)	Distribuição de Jobs Treinamento	Distribuição de Jobs Teste
0	$300 \leq elapsed < 1800$	472030	74450
1	$1800 \leq elapsed < 3600$	148470	14791
2	$3600 \leq elapsed < 7200$	122207	11531
3	$7200 \leq elapsed < 14400$	131702	7904
4	$14400 \leq elapsed < 21600$	48239	5778
5	$21600 \leq elapsed < 28800$	37238	2671
6	$28800 \leq elapsed < 36000$	39225	2425
7	$36000 \leq elapsed < 43200$	43931	1664
8	$43200 \leq elapsed < 50400$	30547	1621
9	$50400 \leq elapsed < 57600$	22320	1440
10	$57600 \leq elapsed < 64800$	16616	1221
11	$64800 \leq elapsed < 72000$	12633	850
12	$72000 \leq elapsed < 79200$	7527	683
13	$79200 \leq elapsed < 86400$	4973	575
14	$elapsed \geq 86400$	36528	3662

Tabela 2. Segundo esquema de discretização do atributo alvo (Classes CESAR).

Classe	Intervalo de Duração (em segundos)	Distribuição de Jobs Treinamento	Distribuição de Jobs Teste
0	$300 \leq elapsed \leq 600$	153811	17474
1	$601 \leq elapsed \leq 1250$	230894	50662
2	$1251 \leq elapsed \leq 4000$	254793	23285
3	$4001 \leq elapsed \leq 6000$	71276	7664
4	$elapsed \geq 6001$	463412	32181

Tabela 3. Desempenho comparativo dos modelos de classificação.

Modelo	Discretização	Acurácia	MAE	MAPE
XGBoost + Target Encoder	Classes CESAR	0,58	11577,62	82,14
J48 + Ordinal Encoder	Classes CESAR	0,62	11562,17	77,91
XGBoost + Target Encoder	Classes UFF	0,68	8642,40	131,58
J48 Two-Step + Target Encoder	Classes UFF	0,64	10451,62	175,19
J48 + Ordinal Encoder	Classes UFF	0,53	10240,45	235,87

Em contrapartida, o modelo **J48 + Ordinal Encoder + Classes CESAR** alcançou o menor MAPE (77,91%), significativamente inferior aos modelos com Classes UFF. Isto sugere que a discretização CESAR, com menos intervalos e mais amplos, oferece maior robustez para a minimização do erro percentual. A acurácia (0,62) e o MAE (≈ 11560) permaneceram altos, indicando que a perda de granularidade inerente a este esquema implica em uma maior incerteza absoluta na previsão.

A estratégia **J48 Two-Step**, que mostrou-se vantajosa em [Nunes et al. 2025] para mitigar o desbalanceamento de classes, não superou o desempenho do XGBoost neste contexto, alcançando acurácia intermediária (0,64) mas com MAE e MAPE elevados. Este resultado pode ser atribuído à interação específica entre a estratégia *Two-Step* e os atributos codificados com *Target Encoder*.

O pior desempenho geral, especialmente em termos de MAPE (235,87%), foi observado no modelo **J48 + Ordinal Encoder + Classes UFF**, destacando a sensibilidade

do algoritmo J48 ao severo desbalanceamento do primeiro esquema de discretização.

2.2. Análise dos Resultados: Conjunto de Teste Por Semana

As Tabelas 4 e 5 apresentam o comparativo das métricas de Acurácia e MAPE por semana, permitindo uma visualização mais clara do desempenho dos diferentes modelos ao longo das cinco semanas de avaliação. Pode-se observar que há uma variação significativa no desempenho para todos os modelos, indicando que fatores temporais e a composição específica de *jobs* em cada semana têm um impacto crucial na qualidade das previsões.

Tabela 4. Resultados de Acurácia, por semana do conjunto de teste.

Modelo	Discretização	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5
XGBoost + Target Encoder	Classes CESAR	0,56	0,58	0,72	0,51	0,62
J48 + Ordinal Encoder	Classes CESAR	0,71	0,60	0,65	0,46	0,52
XGBoost + Target Encoder	Classes UFF	0,82	0,67	0,64	0,40	0,62
J48 Two-Step + Target Encoder	Classes UFF	0,78	0,64	0,64	0,32	0,61
J48 + Ordinal Encoder	Classes UFF	0,67	0,57	0,43	0,40	0,35

Tabela 5. Resultados de MAPE (%), por semana do conjunto de teste.

Modelo	Discretização	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5
XGBoost + Target Encoder	Classes CESAR	56,24	79,53	66,51	96,37	62,18
J48 + Ordinal Encoder	Classes CESAR	55,13	82,41	71,96	113,20	113,86
XGBoost + Target Encoder	Classes UFF	111,22	135,15	120,46	177,24	149,0
J48 Two-Step + Target Encoder	Classes UFF	121,10	146,50	121,79	190,55	116,41
J48 + Ordinal Encoder	Classes UFF	171,58	272,45	158,24	292,93	588,19

Em relação à acurácia, o modelo **XGBoost + Target Encoder + Classes UFF** apresentou a maior amplitude de variação, oscilando entre 82% de acurácia na primeira semana e 40% na quarta semana, embora tenha se mostrado o mais preciso no geral. Em contraste, o modelo **J48 + Ordinal Encoder + Classes UFF** demonstrou ser o menos robusto, com uma queda acentuada a partir da terceira semana, atingindo apenas 35% de acurácia na quinta semana, confirmando sua sensibilidade ao desbalanceamento de classes. Por outro lado, o modelo **J48 + Ordinal Encoder + Classes CESAR** mostrou-se o mais consistente, mantendo desempenho estável sem picos extremos, posicionando-se como opção confiável para cenários que priorizam previsibilidade.

Em relação ao MAPE, o modelo **XGBoost + Target Encoder + Classes UFF**, que obteve alta acurácia nas primeiras semanas, registrou simultaneamente os piores MAPEs (111,22% e 135,15%), demonstrando que erros em classes de *jobs* longos resultam em elevados erros percentuais. Destaca-se como superior o modelo **XGBoost + Target Encoder + Classes CESAR**, que alcançou o menor MAPE em 4 das 5 semanas, mantendo valores consistentemente abaixo de 100%. Este desempenho confirma a maior eficácia da discretização CESAR com classes mais amplas no controle do erro relativo. Em contrapartida, o modelo **J48 + Ordinal Encoder + Classes UFF** apresentou os piores resultados, com MAPE catastrófico de 588,19% na quinta semana, evidenciando previsões extremamente imprecisas para *jobs* longos.

Nota-se ainda que a discretização **Classes CESAR** conferiu maior estabilidade ao MAPE tanto para o XGBoost quanto para o J48, proporcionando previsões mais robustas e confiáveis em termos de erro percentual, mesmo frente a flutuações na acurácia.

3. Conclusão e Trabalhos Futuros

A escolha do esquema de discretização constitui um fator crítico no desempenho de modelos classificadores para a previsão de tempo de execução de *jobs* em ambientes HPC. A análise temporal conduzida neste trabalho demonstrou que, enquanto abordagens com maior granularidade podem alcançar picos pontuais de acurácia, seu desempenho mostra-se volátil e produz erros percentuais inaceitavelmente altos em semanas com distribuição atípica de *jobs*.

A abordagem de discretização com menor granularidade proporcionou robustez superior, mantendo o erro percentual médio em patamares significativamente mais baixos e estáveis ao longo do tempo. O modelo **XGBoost + Target Encoder + Classes CESAR** emergiu como o mais equilibrado, visando cenários reais de implantação, onde previsões consistentemente ruins são mais prejudiciais que previsões ocasionalmente precisas.

Como direções futuras, planejamos (1) investigar métodos avançados de平衡amento por classe; (2) analisar detalhadamente erros por classe; (3) estudar abordagens adaptativas de discretização conforme a distribuição temporal dos *jobs*; e (4) desenvolver modelos de regressão robustos.

Referências

- Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.
- Coats, K. H. (1982). Reservoir Simulation: State of the Art. *Journal of Petroleum Technology*, 34(8):1633–1642.
- Kuchnik, M. et al. (2019). This is why ML-driven cluster scheduling remains widely impractical. Technical report, Carnegie Mellon University.
- Lima, M. et al. (2024). Modelos de Previsão do Tempo de Jobs Aplicados a um Ambiente de Produção de Alto Desempenho. In *Anais do XXV Simpósio em Sistemas Computacionais de Alto Desempenho*, pages 25–36, Porto Alegre, RS, Brasil. SBC.
- Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2018). *Foundations of Machine Learning*. MIT Press.
- Nunes, A. L. et al. (2023). Prediction of Reservoir Simulation Jobs Times Using a Real-World SLURM Log. In *Anais do XXIV Simpósio em Sistemas Computacionais de Alto Desempenho*, pages 49–60. SBC.
- Nunes, A. L. et al. (2024). A Framework for Executing Long Simulation Jobs Cheaply in the Cloud. In *2024 IEEE International Conference on Cloud Engineering (IC2E)*, pages 233–244. IEEE.
- Nunes, A. L. et al. (2025). Two-Step Estimation Strategy for Predicting Petroleum Reservoir Simulation Jobs Runtime on an HPC Cluster. *Concurrency and Computation: Practice and Experience*, 37(4-5):e70026.
- Portella, F., Buchaca, D., Rodrigues, J. R., and Berral, J. L. (2022). TunaOil: A tuning algorithm strategy for reservoir simulation workloads. *Journal of Comput. Science*, 63.
- Quinlan, J. R. (2014). *C4.5: Programs for Machine Learning*. Elsevier.