Adaptação do algoritmo LAX – Lean Adaptive XGBoost para classificação semi-supervisionada de fluxos de dados

Vinícius Giroti, Fabiano Baldo

INTRODUÇÃO

O aumento na utilização de tecnologias de geração, aquisição e comunicação de dados implicou na popularização dos chamados fluxos de dados. Eles se caracterizam pela geração contínua de dados de maneira ilimitada, cujas propriedades estáticas de seus dados variam ao longo do tempo, o que os caracterizam como não estacionários (Barros e Santos, 2019). Dentre os algoritmos de aprendizado de máquina propostos para a análise de fluxos de dados se destaca o SSAFXGB-KNN (Policarpo, 2024). Ele é uma adaptação do algoritmo XGBoost (Chen *et al.*, 2015) para torná-lo apto a processar fluxo de dados. Ele foi escolhido por possuir um processamento rápido e uma boa precisão. Entretanto, em sua implementação, o SSAFXGB-KNN utilizou uma biblioteca Python chamada *scikit-multiflow* (Montiel *et al*., 2018), que foi descontinuada nos últimos anos. Portanto, este trabalho tem como objetivo a refatoração do código do algoritmo SSAFXGB-KNN para substituir a biblioteca *scikit-multiflow* pela biblioteca que a sucedeu chamada *River* (Montiel *et al*., 2021).

DESENVOLVIMENTO

O trabalho consistiu na substituição da biblioteca *scikit-muliflow* (Montiel *et al*., 2018), a qual foi descontinuada, pela biblioteca *River* (Montiel *et al*., 2021), além de realizar experimentos para avaliar os impactos causados no desempenho e precisão do algoritmo após a refatoração. Como a biblioteca River (Montiel *et al*., 2021) surgiu a partir da junção da *scikit-muliflow* (Montiel *et al*., 2018) e da *Creme* (Halford *et al.,* 2019), a substituição de parte dos métodos ocorreu de forma direta, apenas necessitando corrigir seus nomes e parâmetros. Contudo, houve uma mudança significativa na forma de avaliação do modelo devido a não existência do método *"Evaluate Prequential*" na biblioteca *River*. Para contornar este problema, foram implementadas duas formas de avaliação do modelo, sendo a primeira por meio da implementação de um método com lógica semelhante a lógica do "*Evaluate Prequential*", e a segunda utilizando o *framework Maki Nage* (Picard, 2020), gerando duas implementações do algoritmo chamadas, SSAFXGB-KNN-River e SSAFXGB-KNN-MakiNage, respectivamente.

RESULTADOS

Os experimentos seguiram o mesmo protocolo e utilizaram os mesmos *datasets* de Policarpo (2024). Como visto na Tabela 1, os algoritmos refatorados tiveram desempenho levemente inferior ao original em todas as métricas de precisão, porém teve um aumento significativo no tempo de execução, cerca de 3 vezes. Isso ocorreu porque os métodos da *River* (Montiel *et al*., 2021) não suportarem processamento de dados em *batches*. Como o SSAFXGB-River e o SSAFXGB-MakiNage usa a mesma lógica de predição, os valores de Acurácia, Kappa e F1 de ambos são iguais, portanto, apenas os resultados de precisão do SSAFXGB-River foram utilizados nos testes estatísticos. Os testes estatísticos de Friedman apontaram significância estatística nos resultados com confiança de 95%. A Figura 1 mostra que o teste de Nemenyi identificou dois grupos de algoritmos com desempenho semelhante, onde em ambos o algoritmo original e os refatorados estão contidos, mostrando que apenas de terem um desempenho inferior, isso não se demonstram ser expressivo.

CONSIDERAÇÕES FINAIS

As adaptações ao algoritmo SSAFXGB causaram queda no desempenho na maioria dos experimentos, entretanto, essa queda não foi expressiva em termos de Acurácia, Kappa e F1. Em contrapartida, no quesito tempo de execução, o aumento foi significativamente maior, chegando a 3 vezes o tempo do algoritmo original. Portanto, como conclusão, é possível perceber que a refatoração foi consistente, pois manteve os resultados de precisão, mas com prejuízo no tempo de execução.

Palavras-chave: refatoração, fluxo de dados, semi-supervisionado, dados não estacionários, multi-classe, xgboost.

ILUSTRAÇÕES

Tabela 1. *Médias de acurácia, Kappa, F1 e tempo em segundos dos modelos.*

	SSAFXGB-KNN-Original			SSAFXGB-KNN-River			SSAFXGB-KNN- MakiNage		
% de Dados Não Rotulados	0,9	0,95	0,99	0,9	0,95	0,99	0,9	0,95	0,99
Acurácia	0,87	0,84	0,73	0,85	0,81	0,64	0,85	0,81	0,64
Kappa	0,81	0,77	0,57	0,78	0,73	0,47	0,78	0,73	0,47
F1	0,75	0,73	0,63	0,74	0,71	0,54	0,74	0,71	0,54
Tempo (seg)	274,26	280,07	296,21	850,1	862,36	868,49	818,87	826,25	844,65

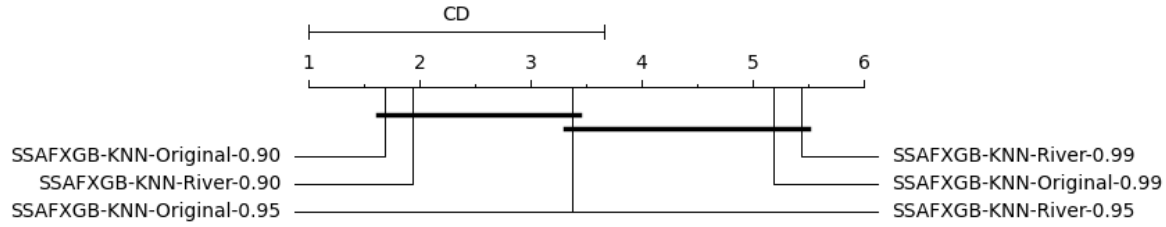

Figura 1. *Teste de Nemenyi para Acurácia entre os modelos Orignais e Refatorados usando River.*

REFERÊNCIAS BIBLIOGRÁFICAS

BARROS, R. S. M. de; SANTOS, S. G. T. de C. An overview and comprehensive comparison of ensembles for concept drift**. Information Fusion**, v. 52, p. 213–244, 2019. ISSN 1566-2535. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1566253518308066>.

CHEN, T. et al. Xgboost: extreme gradient boosting. **R package version** 0.4-2, v. 1, n. 4, p. 1–4, 2015.

HALFORD, M.; ET AL. **Creme, a Python library for online machine learning**. 2019. Disponível em: https://github.com/MaxHalford/creme. Acesso em: 16 ago. 2025.

MONTIEL, J.; READ, J.; BIFET, A.; ABDESSALEM, T. Scikit-multiflow: A multi-output streaming framework. **The Journal of Machine Learning Research**, 19, 72, p. 1-5, 2018. Disponível em: http://jmlr.org/papers/v19/18-251.html. Acesso em: 16 ago. 2025.

MONTIEL, J.; ET AL. River: Machine Learning for streaming data in Python. **The Journal of Machine Learning Research**, 22, 110, p. 1-8, 2021. Disponível em: http://jmlr.org/papers/v22/20-1380.html. Acesso em: 16 ago. 2025.

PICARD, R. **Maki Nage:** Stream Processing made Easy. 2021. Disponível em: https://www.makinage.org/. Acesso em: 16 ago. 2025.

POLICARPO, D. A. **SSAFXGB:** Classificação semi-supervisionada multi-classe de fluxo de dados com a utilização do XGBOOST. 2024. Mestrado em computação aplicada – Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas, 24 maio 2024.

DADOS CADASTRAIS

BOLSISTA: Vinícius Giroti
MODALIDADE DE BOLSA: PROBIC/UDESC (IC)
VIGÊNCIA: 09/2024 a 08/2024 – Total: 12 meses
ORIENTADOR(A): Fabiano Baldo
CENTRO DE ENSINO: CCT
DEPARTAMENTO: DEPARTAMENTO DE CIENCIA DA COMPUTACAO CCT
ÁREAS DE CONHECIMENTO: Ciências Exatas e da Terra/ Ciência da Computação
TÍTULO DO PROJETO DE PESQUISA: StreamMining v.3.0 Novas Abordagens para Algoritmos de Aprendizagem em Fluxos de Dados Não Estacionários
Nº PROTOCOLO DO PROJETO DE PESQUISA: NPP3216-2023