Universidade Federal do Paraná
Setor de Ciências Exatas
Departamento de Estatística
Programa de Especialização em *Data Science* e *Big Data*

Luiz Gabriel de Souza

Automated Machine Learning for classification problems: Comparative analysis of private and open-source solutions

Curitiba
2022

Luiz Gabriel de Souza

Automated Machine Learning for classification problems: Comparative analysis of private and open-source solutions

Monografia apresentada ao Programa de Especialização em Data Science e Big Data da Universidade Federal do Paraná como requisito parcial para a obtenção do grau de especialista.

Orientador: Ph.D. Luiz Eduardo S. Oliveira

Curitiba

2022

Especialização em Data Science e Big Data
Universidade Federal do Paraná
dsbd.leg.ufpr.br

Automated Machine Learning for classification problems: Comparative analysis of private and open-source solutions

Luiz Gabriel de Souza[1]
Ph.D. Luiz Eduardo S. Oliveira[2]
Outro Orientador do Programa[3]
Orientador Externo[4]

Resumo

O crescente uso de soluções baseadas em machine learning para resolução de problemas complexos em diversas áreas de aplicação, tem demandado soluções de aprendizado de máquina versáteis e de fácil utilização, que não demandem muitas habilidades técnincas para a sua execução. A partir de meados da década passado, começaram a surgir soluções voltadas para esse objetivo, denominadas *Automated Machine Learning* (AutoML), com iniciativas da comunidade de desevolvimento de códigos abertos e de grandes empresas privadas. Com base nessa realidade, diversos projetos tem buscado comparar a performance de *frameworks* de AutoML, testando diferentes bases de dados, arquiteturas e parametrizações aceitas por cada *framework* Este projeto tem como objetivo comparar a performance de seis *frameworks* de AutoML escolhidos, sendo três deles pagos (Google Auto Machine Learning, Amazon Web Services Autopilot e Dataiku) e três de código aberto (Pycaret, MLJar e H20), para modelos de classificação binários, visando entender se há diferenças consideráveis entre a parformance dos *frameworks* pagos e gratuitos de AutoML. O projeto fez uso de 10 bases de dados públicas, que foram processadas com o menor ajuste possíveis de parametrização permitidos por cada framework *frameworks*, buscando emular o uso de usuários com pouca ou nenhuma experiência em machine learning. ADICIONAR CONCLUSÔES
Palavras-chave: Automate Machine Learning, AutoML, Pycaret, GCP, Autopilot.

Abstract

The growing use of suitable solutions in machine learning to solve complex problems in several areas, has demanded versatile machine learning solutions, which do not require many technical skills for their execution. From the entire decade, solutions began to emerge from the community of automated objectives, called Machine Learning (AutoML), with open source and large market initiatives. Based on this reality, several projects have sought to compare the performance of AutoML frameworks, testing different databases, architectures and parameterizations accepted by each framework. This project aims to compare the performance of six chosen AutoML frameworks, three of which are paid (Google Auto Machine Learning, Amazon Web Services Autopilot and Dataiku) and three are open source (Pycaret, MLJar and H20), for binary classification models. , understand if there are considerable differences between the performance of the paid and free AutoML frameworks. The project made use of 10 public databases, which were processed with the least possible parameterization allowed by each framework frameworks, seeking to emulate the use of adjustments from users with little or no experience in machine learning. ADICIONAR CONCLUSÔES
***Keywrods**: Automate Machine Learning, AutoML, Pycaret, GCP, Autopilot.*

1 Introdução

1.1 Crescimento do uso de Machine Learning

As últimas décadas marcaram o crescimento do uso de soluções de machine learning(ML) por grandes empresas e governos no mundo todo. Foram criadas soluções robustas para resolver problemas de visão computacional, reconhecimento de fala, jogos, classificação de elementos, agrupamentos, entre outras. É possível encontrar aplicações de ML na indústria com máquinas interconectadas que prevem problemas de produção; em hospitais auxiliando no diagnósticos de paciêntes; nas escolas e universidades personalizando o ensino de estudos; no marketing das empresas, personalizando e maximizando os resultados de campanhas de comunicações; etc.

Com essa popularização, o interesse de profissionais que buscam os benefícios dessa tecnologia também cresceu, fazendo com que a presença e o valor de soluções que exijam pouco ou nenhum conhecimento prévio so-

[1]Aluno do programa de Especialização em Data Science & Big Data, luiz.gabriel.souza@hotmail.com.
[2]Professor do Departamento de Estatística - DEST/UFPR, luiz.oliveira@ufpr.br.
[3]Professor do Departamento de Informática - DINF/UFPR.
[4]Chefe do Departamento de Data Science.

bre machine learning tenha se expandido. Empresas privadas e a comunidade de desenvolvedores de código aberto passou a criar soluções que automatizam o processo de experimentação que é feita normalmente por experiêntes profissionais das áreas de estatística e matemática. Grandes empresas de tecnologia como Amazon e Google, por exemplo, criaram soluções deste caráter, onde com poucos cliques em uma plataforma de cloud interativa, é possível treinar modelos complexos e extrair resultados que podem ser utilizados imediatamente, sem a necessidades de configuração prévia de todo o fluxo tradicional de construção de um modelo de ML. Ao mesmo tempo, iniciativas de código aberto criaram ferramentas como Pycaret [1] e MLJar [2], que apesar de não terem interface interativa como as citadas pelas empresas privadas acima, conseguem entregar soluções complexas e realizar diversos tipos de experimentos e transformação de dados com poucas linhas de código, de maneira intuitiva e rápida, sendo acessível para novos usuários interessados em modelos de ML e garantindo boa performance.

Além de atendar aos interesses de profissionais com pouca experiência no desenvolvimento de soluções de ML, *frameworks* de AutoML podem contribuir facilidando o dia a dia de profissionais experientes, automatizando processos de experimentação e otimização de parâmetros, economizando tempo e aumentando a produtividade dos especialistas na área: *"Put simply, AutoML can lead to improved performance while saving substantial amounts of time and money, as machine learning experts are both hard to find and expensive. As a result, commercial interest in AutoML has grown dramatically in recent years, and several major tech companies are now developing their own AutoML systems."* [3]

1.2 Objetivo do projeto

Com base no crescimento do uso de soluções de AutoML, tanto privadas quanto abertas, este projeto tem como objetivo comparar a performance de *frameworks* de AutoML para problemas de classificação binários. Não serão realizados nenhum tipo de pré processamento ou engenharia de caracteríticas dos dados, visando testar as funcionalidades em cada *framework*. Além disso, tentaremos emular o uso dos frameworks por usuários com pouca experiência em ML, fazendo uso do maior uso de parâmetros padrão de cada *framework*

2 Dados

A fim de a garantir comparabilidade entre os resultados dos testes e ao mesmo tempo considerar as dificuldades de se lidar com dados categóricos e numéricos, selecionamos dez *datasets* disponibilizadas pelo projeto aberto OpenML [4], utilizadas para a criação do AutoML Benchmark Framework [5], idealizado pelo mesmo. Na documentação do projeto [5], a escolha dos *datasets* foi descrita como: *"The benchmark aims to consist of datasets that represent real-world data science problems. This means we want to include datasets of all sizes (including big ones), of different problem domains and with various levels of difficulty"*.

Dentre todas os *datasets* disponíveis, foram escolhidos os que possuiam diferentes volumes de atributos e instancias entre si, bem como diferentes tipos de dados como apenas categóricos, apenas numérico e numéricos e categóricos ao mesmo tempo. O nome e algumas características dos *datasets* escolhidos pode ser observado abaixo:

\begin{table}
\caption{(#tab:table_datasets)Data-sets selecionados}

Nome	Atributos	Instancias	Tipo.de.Atributos
KDDCup09_appetency	231	50.000	Mix
jasmine	145	2.984	Mix
nomao	119	34.465	Mix
kr-vs-kp	37	3.196	Categórico
higgs	29	98.050	Numérico
kc1	22	2.109	Numérico
bank-marketing	17	45.211	Mix
adult	15	48.842	Mix
Amazon_employee_access	10	32.769	Categórico
blood-transfusion	5	748.000	Numérico

\end{table}

Todos os *datasets* podem ser encontrado neste link

3 Experimentos

Foram testados seis *frameworks* de autoML, três deles privados e outros três de código aberto. Entre os privados, foram testados o Google Cloud AutoML (GCP AutoML) GCPAutoML, Amazon AutoPilot (AWS AutoPilot) [6] e Dataiku [7]. Entre os de código aberto, foram testados PyCaret [1], MLJar [2] e H2O [8].
Para garantir a comparabilidade dos experimentos, algumas premissas foram tomadas em conta para todos os modelos processados:

- **Tempo de Processamento:** Limitado a 60 minutos o tempo de processamento para o treino de cada *datasets* em cada *framework*.
- **Ambiente de Processamento:** Para os *frameworks* privados, foram utilizados seus respectivos ambientes de processamento em nuvem sem personalização. Para os de código aberto, utilizamos a plataforma Google Colab com suas configuações padrões do pacote gratúito. Os scripts podem ser acessados aqui.
- **Métrica para otimizar:** Quando permitido pelo *framework*, optamos por otimizar os experimentos para a métrica **ROC-AUC**, escolhida como medida de comparação entre os resultados obtidos de cada framework - a mesma utilizada pelo projeto [5] para problema de classificação binária.
- **Paramêtros opcionais:** Optamos por utilizar a maior quantidade de parâmetros padrãos de cada framework, ou seja, sem utilizar parâmetros como balanceamento automático de variáveis, quantidade de modelos para treino, etc. Adicionamos, quando aplicável, a semente aleatória igual a 123, buscando reproducibilidade dos experimentos.
- **Outras limitações intencionais:** Foram desconsirados resultados de modelos *Ensembles* e Redes Neurais, visto que nem todas os *frameworks*possuem tais *features*.

3.1 *Frameworks* Privados

Com o aumento da popularidade de soluções em cloud, que geralmente são pagas, as empresas investiram também em soluções de machine learning pré prontas para uso, como é o caso do Google Cloud Plataform (GCP), Amazon Web Services (AWS) e a Dataiko. Abaixo alguns comentário sobre nossa experiência de usuário sobre cada uma delas:

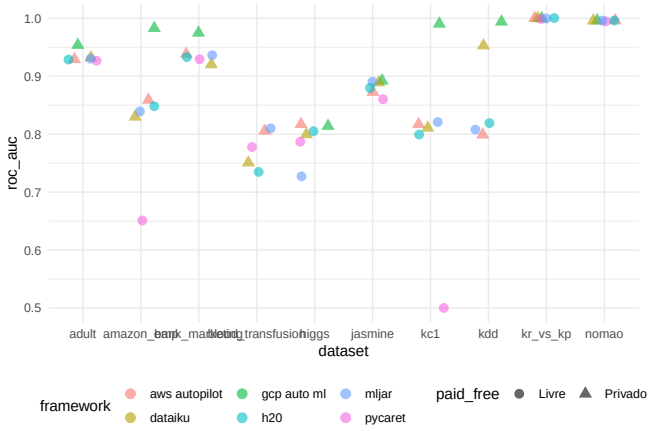
- **AWS Sagemaker AutoPilot**
- **Google Cloud Plattaform Auto Machine Learning**
- **Dataiku** Plataforma clara e intuitiva. Basta importar o *dataset*

3.2 *Frameworks* Gratuitos

- **Pycaret** Se demonstrou uma biblioteca de fácil utilização, com muitos recursos ajustáveis e fácil acesso aos objetivos criados durante os testes como as bases transformadas, métricas, entre outros. , fazendo dela uma boa ferramenta tanto para
  utilizada Sklearn como base, o que facilita demais, mas não se limita apenas a modelos existentes no Sklearn. Possui compatibilidade para a adição de outros modelos, inclusive Redes Neurais, e suporte para processamento com GPUs.
- **MLJar**
- **H20 (Versão Gratúita)** Possui arquitetura própria, sendo necessário a tranformação dos dados de treino para o formato exclusive da biblioteca. Além disso

4 Resultados e discussão

No gráfico @ref(fig:resultados_auc) é possível visualizar os resultados comparatívos da métrica ROC-AUC apra cada um dos *datasets* em cada *frameworks*. Vemos que ...



5 Conclusão

Aqui
Importante ressaltar que não os resultados para cada framework não passaram por testes posteriores para analisar a presença de overfitting ou qualquer outro problema de generalização que os resultados possam ter.
Como próximos passos acredito ser relevante estruturar uma maneira efetiva e comparável de coletar dados de custos de processamento em ambiente Cloud.

Referências

[1] Moez Ali. *PyCaret: An open source, low-code machine learning library in Python*, April 2020. PyCaret version 1.0.0.

[2] Aleksandra Płońska and Piotr Płoński. Mljar: State-of-the-art automated machine learning framework for tabular data. version 0.10.3, 2021.

[3] Joaquin Vanschoren Frank Hutter, Lars Kotthoff. *Automated Machine Learning*. 2019.

[4] Bernd Bischl Joaquin Vanschoren, Jan N van Rijn. OpenML: networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013.

[5] P. Gijsbers, E. LeDell, S. Poirier, J. Thomas, B. Bischl, and J. Vanschoren. An Open Source AutoML Benchmark. *arXiv preprint arXiv:1907.00909 [cs.LG]*, 2019. Accepted at AutoML Workshop at ICML 2019.

[6] Amazon Web Services. *Amazon SageMaker Developer Guide*.

[7] Dataiku. *Dataiku Data Science Studio*.

[8] Erin LeDell and Sebastien Poirier. H2O AutoML: Scalable automatic machine learning. *7th ICML Workshop on Automated Machine Learning (AutoML)*, July 2020.