

Aprendizado de Máquina Automático (AutoML) para problemas de classificação: Análise comparativa do desempenho de frameworks privados e públicos.

Luiz Gabriel de Souza

Orientador: Ph.D. Luiz Eduardo S. Oliveira

Universidade Federal do Paraná

Setor de Ciências Exatas

Departamento de Estatística

Programa de Especialização em Data Science e Big Data

25 de Junho de 2022

Automated Machine Learning

Automated Machine Learning

- Auto ML fornece métodos e processos para disponibilizar o Aprendizado de Máquina para não especialistas na área, para melhorar a eficiência e acelerar o processo de experimentação de criação de modelos.
- Frameworks de AutoML podem ser capazes de:
 - ▶ Pré-processamento e limpeza dos dados;
 - ▶ Selecionar e construir vetores de características;
 - ▶ Selecionar uma família de modelos apropriados para o problema;
 - ▶ Otimizar hiperparâmetros do modelo;
 - ▶ Construir a topologia de redes neurais;
 - ▶ Analisar os resultados obtidos.

Literatura

- O projeto *AutoML Benchmark Framework* da Fundação Open Machine Learning (openml.org) serviu como base para parte da metodologia adotada nos experimentos, uma vez que o benchmark de AutoML é o mais robusto que encontramos na literatura;
- Não foram identificados artigos ou outros benchmarks robustos que analisem a performance de frameworks de AutoML privados.

Objetivo do Projeto

Objetivo do Projeto

- O objetivo do projeto é comparar as características e a performance de frameworks de Auto ML privados e públicos;
- Os frameworks escolhidos foram:

Privados



Públicos



Experimentos

Dados

- Foram escolhidos 10 datasets também utilizados pelo projeto de referência *OpenML Benchmark Framework*;
- Os datasets escolhidos possuem diferentes volumes de atributos e instâncias entre si, bem como diferentes tipos de dados;

Tabela 1: Data-sets selecionados

Nome	Atributos	Instancias	Tipo_Atributos
kdd	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_emp	10	32.769	Categórico
blood_transfusion	5	748.000	Numérico

Premissas

- Tempo de processamento: 1 hora;
- Ambiente de Processamento: Cloud para privados e Google Colab para públicos;
- Métrica de otimização: ROC-AUC;
- Parâmetros adicionais: Optamos por utilizar a maior quantidade possível de parâmetros padrão de cada ferramenta;
- Outras limitações: Quando possível, desconsideramos resultados de Redes Neurais e modelos Ensemble.

Experiência de uso

Privados

- Interface interativa de fácil uso;
- Falta de clareza quanto aos custos durante processamento;
- AWS: Exporta os notebooks criados automaticamente pela ferramenta;
- Google: Não mostra metodologia, modelo utilizado ou qualquer tipo de informação sobre o processamento dos dados;
- Dataiku: O mais intuitivo e fácil de usar.

Públicos

- Necessidade de conhecimentos de código em python para uso;
- Dificuldades em processar datasets muito grandes por limitação de máquina;
- H2O: Necessidade de configuração próprias da biblioteca como a transformação de DataFrames;
- MLJar: Exporta um HTML interativo com os resultados dos experimentos, incluindo gráficos para análise;
- Pycaret: O mais intuitivo e fácil de usar, pois, precisa de poucas linhas de código.

Resultados

Comparação ROC-AUC

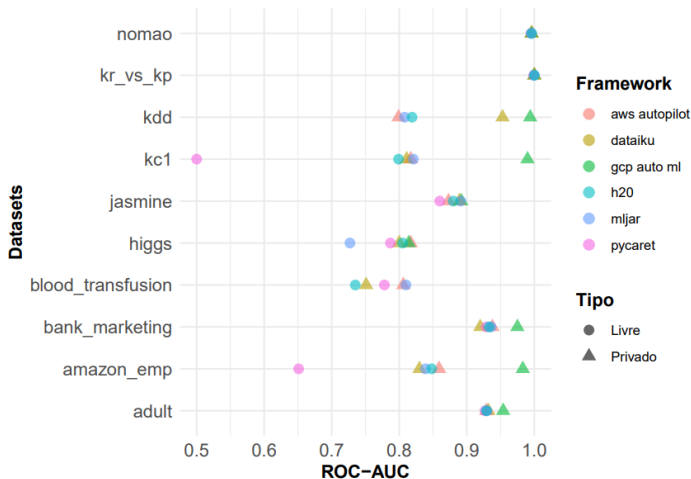


Figura 1: ROC-AUC Classificação Binária

Ranking por Framework

- Para analisar a performance dos frameworks de forma agrupada, utilizamos o seguinte método:
 - ▶ Para cada dataset, criamos um ranking ordenado pela performance da métrica ROC-AUC de cada framework, onde o melhor desempenho fica na posição 6 e o pior desempenho recebe a posição 1;
 - ▶ Dessa forma, o ranking passa a valer como uma espécie de pontuação com base no desempenho de cada framework, possibilitando a criação dos seguintes gráficos:

Ranking por framework

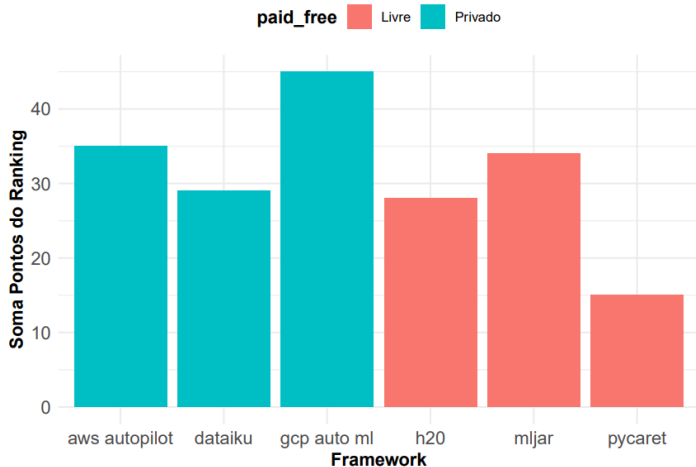


Figura 2: Soma de pontos por Framework

Ranking por tipo

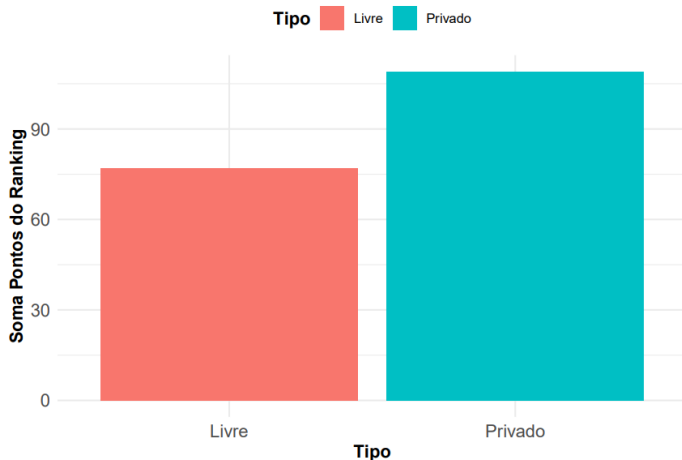


Figura 3: Soma de pontos por tipo de Framework

Conclusões

Conclusões

- O framework Google obteve a melhor performance mas é uma “caixa preta” e não diz qual tipo de técnica utilizou para alcançar seus resultados;
- Comparando os demais frameworks, os públicos tiveram resultados muito competitivos quando comparados com os privados, com destaque para o MLJar que obteve resultados muito próximos aos da AWS Autopilot;
- Pycaret obteve a pior performance entre todos os frameworks analisados.

Possíveis próximos passos

- Coletar dados de custos financeiros das plataformas privadas;
- Utilizar todas as features de cada framework, incluindo Deep Learning;
- Estender a quantidade de datasets, incluindo problemas de classificação multi classe e regressão.

Obrigado!