Benchmarking de Sistemas AutoML Open-source

Open-source AutoML Systems Benchmarking

Maria Santos[1]
orcid.org/0000-0002-3590-2704

Gabriel Mac'Hamilton[2]
orcid.org/0000-0002-3735-190X

Alexandre Maciel[3]
orcid.org/0000-0003-4348-9291

[1]Escola Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil. E-mail: mvrs2@ecom,p.poli.br

DOI: 10.25286/repa.v7i3.2456

Como citar este artigo pela NBR 6023/2018: Maria Santos; Gabriel Mac'Hamilton; Alexandre Maciel. Benchmarking de Sistemas AutoML Open-source. Revista de Engenharia e Pesquisa Aplicada, Recife, v. 7, n. 3, p. 19-28.

RESUMO

Este estudo propõe comparar três sistemas *AutoML* (Aprendizado de Máquina Automatizado) de código aberto mais conhecidos, o *Auto*-WEKA, *Auto-Sklearn* e *TPOT*, em termos de funcionamento em cada parte do fluxo de um *AutoML,* e algoritmos suportados em cada parte desse fluxo. O Aprendizado de Máquina Automatizado é uma ferramenta que automatiza o resultado do aprendizado de máquina com o mínimo de esforço humano possível. Este trabalho mostra que, para determinados tipos de dados e objetivos de previsão, o usuário, seja estudante ou profissional da área de ciência de dados, deve se atentar a cada ferramenta, pois implica nos resultados obtidos e as predições podem ficar mais refinadas ou não.

PALAVRAS-CHAVE: Aprendizado de Máquina Automatizado; AutoML; Machine Learning; Auto-WEKA; Auto-Sklearn; TPOT;

ABSTACT

This work offers a benchmarking of the three most know open-source AutoML systems, Auto-WEKA, Auto-Sklearn and TPOT, in terms of how they work in each part of an AutoML flow, and algorithms supported in each part of that flow. Automated Machine Learning is a tool that automates the result of machine learning with as little human effort as possible. This work shows that for certain types of data and forecasting objectives, the users, whether data science students or professionals, must pay attention to each tool, because it implies the results obtained and predictions can be more refined or not.

KEY-WORDS: Automated Machine Learning; AutoML; machine Learning; Auto-WEKA; Auto-Sklearn; TPOT;

1 INTRODUÇÃO

Nesta seção será abordada a contextualização do estudo, juntamente com seus objetivos e justificativa.

1.1 CONTEXTUALIZAÇÃO

Meios de comunicação tecnológicos amplamente utilizados no nosso cotidiano, produzem uma vasta massa de dados a todo momento, que pode ser vinda de um simples sistema interno de uma empresa, de um site ou de várias redes sociais, por exemplo, estima-se que são gerados por dia mais de 2^{18} *bytes* **[1]**, esse grande volume é praticamente inviável de ser manipulado apenas por um ser humano. Para isso, foram desenvolvidas soluções a partir da inteligência artificial para resolver problemas de manipulação dos dados, uma delas foi chamada de *Machine Learning* (Aprendizado de Máquina) **[2]**.

Aprendizado de Máquina é uma área de Inteligência Artificial **[3]** cada vez mais presente no dia a dia de um cientista de dados, seus processos possuem algumas características que tornam eles complexos, possuem um alto custo humano, desde o desenvolvimento de código até a fase de testes.

O *AutoML* (Aprendizado de Máquina Automatizado) tem como objetivo automatizar os resultados, reduzindo o esforço humano para aplicar o Aprendizado de Máquina, melhorar a performance computacional, em termos de custo computacional, desempenho dos algoritmos, tempo que é levado para obter os resultados. O modelo possui métricas para melhorar o refinamento dos resultados, possui resultados mais assertivos, melhorando a produção e imparcialidade dos estudos **[2]**.

Neste trabalho, iremos fazer uma análise de sistemas *AutoML open-source* (código aberto) mais utilizados e conhecidos pela comunidade de ciência de dados **[4]**, considerando três deles, baseado nas condições de comparar o primeiro sistema AutoML *open-source* desenvolvido **[5]**, com dois sistemas baseados na biblioteca *Scikit-Learn* da *API Python* que utilizam diferentes algoritmos de previsão **[6]**. São eles: *Auto-WEKA*, *Auto-Sklearn* e o *TPOT.*

1.2 OBJETIVOS

Esse estudo tem como objetivo realizar um *benchmarking*, classificando sistemas de *AutoML open-source*, resultando num auxílio profissionais e estudantes da área de ciência de dados. São objetivos específicos: apontar os sistemas mais conhecidos de código aberto de *AutoML*, analisar suas documentações e especificações e classificar esses sistemas através das análises feitas.

1.3 JUSTIFICATIVA

Com o crescimento da ciência de dados, visto que ela é de extrema importância, para guiar predições, tomadas de decisões de uma empresa ou na área *de* serviço governamental e demais áreas que envolve vida de pessoas e sistemas financeiros, uma boa escolha de ferramentas que auxilia nesse processo é de grande relevância. A escolha de um sistema pode decidir se o resultado daquela previsão está mais refinado e mais assertivo que um outro com ferramenta diferente. Pelo lado colaborativo, o estudo possui o propósito de ajudar estudantes e entusiastas da área de ciência de dados que estão dando seus primeiros passos, em busca de uma ferramenta que melhor se adequa aos seus objetivos.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são explicados conceitos e fundamentações necessárias para o entendimento e melhor compreensão do estudo. O propósito é facilitar o acompanhamento do trabalho, através da apresentação de fundamentos básicos de Aprendizado de Máquina e *AutoML*, além de breve abordagem sobre cada um dos sistemas de Aprendizado de Máquina automatizados analisados.

2.1 APRENDIZADO DE MÁQUINA

O aumento da utilização de meios eletrônicos nas últimas décadas tem gerado uma produção em grande escala de dados, seja de uma simples página de blog até as grandes redes sociais, todos os dados utilizados pelos usuários são rastreados e armazenados a todo momento. Esses dados são inviáveis para tratamentos e análises apenas por seres humanos. Para toda essa manipulação de dados, foram estudados e criados algoritmos que resolvam esse problema de otimização das análises dos dados, desde a coleta, ao armazenamento

deles, e finalmente, aos resultados obtidos de todo o processo, ou seja, o sistema irá rodar o conjunto de dados, processar eles e a partir desse processamento, identificar alguma anomalia, tentar uma aproximação de resultados e fazer as previsões que desejar.

O Aprendizado de Máquina é um método de análise de dados, com pouca interferência humana, faz parte da área da inteligência artificial, e como o próprio nome já diz, o sistema "aprende" com os dados e identifica certos padrões comportamentais e assim toma sua decisão. O algoritmo toma decisões baseadas num resultado anterior bem sucedido **[3]**. Essas "aprendizagens" podem ser de forma supervisionada (em que já se tem uma entrada e saída, a ideia é entender o trajeto desses dados) e não supervisionada (onde há apenas a entrada dos dados, e assim, encontrar os padrões e compreender as correlações entre os dados), a primeira forma pode se dividir em dois métodos (classificação e regressão), já a segunda o método usado para análise é a *clusterização* (dados agrupados por semelhança, a partir da distância entre eles) **[7]**. O Aprendizado de Máquina possui hiperparâmetros que são definidos antes do treino, são valores que otimizam o processo de treinamento do modelo.

Aprendizado de Máquina está muito presente no nosso cotidiano, como por exemplo em serviços de *streaming* com recomendações baseadas no que o usuário assiste, reconhecimento de imagens na área da saúde ou em redes sociais, reconhecimento de padrões textuais, análise e detecção de fraudes, previsão na bolsa de valores e mais inúmeras situações.

2.2 *AutoML*

Todo sistema de Aprendizado de Máquina possui hiperparâmetros, e a tarefa mais básica de um *AutoML* é definir automaticamente esses hiperparâmetros para otimizar o desenvolvimento. Redes neurais profundas recentes, dependem de uma ampla opção de hiperparâmetros, em arquitetura, regularização e otimização da rede neural **[2]**. O *AutoML* possui três principais componentes: a Otimização Automatizada de Hiperparâmetros (*Hyperparameter Optimization*, HPO), o Meta-Aprendizado (Meta-Learning) e o Algoritmo de Busca de Arquitetura de Redes Neurais (*Neural Architecture Search*, NAS) **[8].**

A HPO, utilizada no *AutoML*, é muito mais reprodutiva que a pesquisa manual **[2]**. Por exemplo, em um treinamento com 7 parâmetros {a, b, c, d, e, f, g, h} e cada parâmetro possui 5 valores possíveis, para toda a combinação ser feita, seria necessário treinar 1607 modelos (7^5), o que seria impossível em condições humanas. Com a HPO que otimiza as combinações dos conjuntos de hiperparâmetros, aumentando assim a reprodutibilidade e melhorando o desempenho dos modelos durante os treinamentos, o Aprendizado de Máquina Automatizado (*AutoML*) visa tomar essas decisões de forma orientada por dados, objetiva e automatizada: o usuário simplesmente fornece dados, e o sistema *AutoML* determina automaticamente a abordagem com melhor desempenho para esta aplicação específica **[9]**.

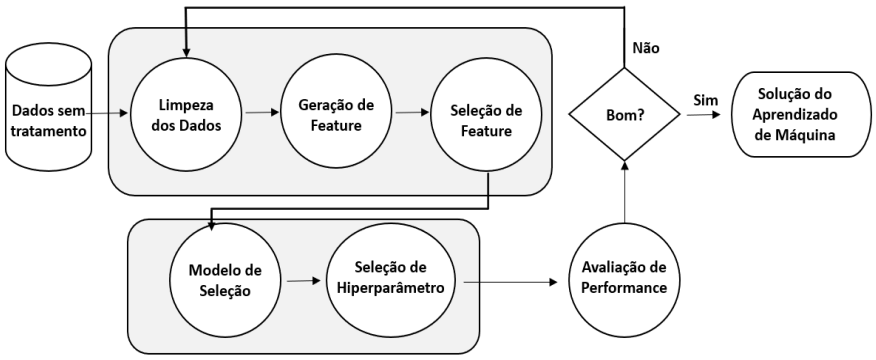
O *AutoML* torna a experiência mais acessível a cientistas de dados que não possuem recursos para aprender sobre as tecnologias atrás dele em detalhes. Isso pode ser visto como uma democratização do Aprendizado de Máquina: com AutoML, Aprendizado de Máquina de última geração personalizado está ao alcance de todos **[10]**.

O *Meta-Learning* (Meta-Aprendizado) é uma técnica que realiza o aprendizado sobre os metadados de vários outros algoritmos de Aprendizado de Máquina **[8]**. O algoritmo vai aprendendo por si só, a cada treinamento o algoritmo vai entendendo ainda mais sobre os dados, o treino anterior é utilizado para otimizar o próximo e assim buscando com mais eficiência o aprendizado.

O algoritmo NAS, é uma técnica para encontrar a melhor arquitetura neural, de acordo com o conjunto de dados que foi enviado a rede neural **[8]**. O NAS possui três componentes importantes: espaço de busca de arquitetura neural, arquitetura de otimização e métodos de modelo de estimação **[11]**. Na Figura 1 é mostrado o esquema de um *AutoML*, desde a preparação dos dados, até o modelo de validação, este ciclo é base para um *AutoML.*

DOI: 10.25286/repa.v7i3.2456

Figura 1 – Esquema de um *AutoML* clássico



Fonte: [12].

2.3 FERRAMENTAS ANALISADAS

Nesta seção serão brevemente apresentados os três sistemas de *AutoML* analisados. São eles: *Auto-WEKA, Auto-Sklearn* e *TPOT*.

2.3.1 *Auto-WEKA*

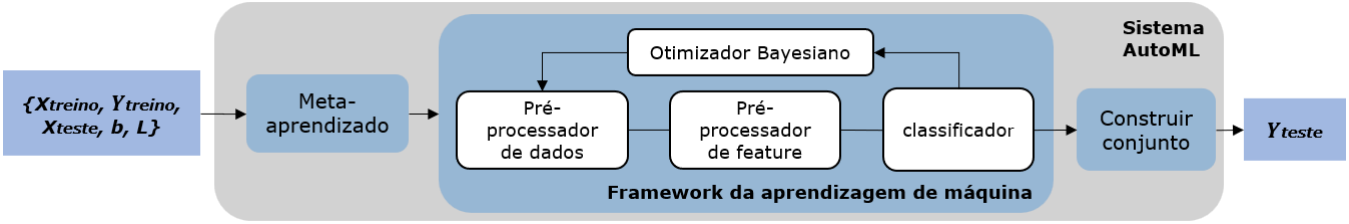
WEKA é uma plataforma de Aprendizado de Máquina de código aberto amplamente utilizada, feita em Java. Devido à sua interface intuitiva, é popular entre usuários iniciantes. No entanto, esses usuários geralmente acham um pouco difícil para identificar a melhor abordagem para seu conjunto de dados específico entre os muitos disponíveis, daí surgiu o *Auto-WEKA* **[13]**. Criado em 2012, é um pacote totalmente integrado ao *WEKA*, um sistema projetado para ajudar esses usuários, automatizando busca criteriosa no espaço conjunto dos algoritmos de aprendizado do *WEKA* e seus respectivos configurações de hiperparâmetros para maximizar o desempenho, usando Otimização Bayesiana **[2] [10]**. Em 2017 foi implementado o *Auto-WEKA* 2.0 **[14]**, o primeiro sistema *AutoML open*-source, adicionando algoritmos de regressão e funcionamento simultâneo, além de considerar a Otimização Bayesiana baseada em árvore, o que resultou em uma eficiência melhor no algoritmo **[15]**. Neste trabalho, *Auto-WEKA* e *Auto-WEKA* 2.0 serão considerados o mesmo sistema.

Dentro do HPO, temos o conceito de CASH (*Combined Algorithm Selection and Hyperparameter Optimization*), que a partir de um conjunto de dados, escolhe automaticamente o algoritmo de aprendizado e paralelamente configura os hiperparâmetros de forma que o desempenho na tarefa seja otimizado **[16]**. O *Auto-WEKA* obteve a primeira aplicação, em que no artigo referente ao sistema, foi apresentado o termo CASH **[14]**.

2.3.2 *Auto-Sklearn*

O *Auto-Sklearn*, criado em 2015, é um sistema baseado na API (*Aplication Programming Interface*, Interface de Programação de Aplicativos) do *Scikit-Learn* (utiliza 15 classificadores, 14 métodos de pré-processamento de *feature* e 4 métodos de pré-processamento de dados, dando origem a um espaço de hipóteses estruturado com 110 hiperparâmetros). O algoritmo melhora em métodos existentes do *AutoML*, levando em consideração automaticamente o desempenho anterior em conjuntos de dados semelhantes e construindo conjuntos a partir dos modelos avaliados durante a otimização **[17]**.

Figura 2 – Esquema do *Auto-Sklearn*



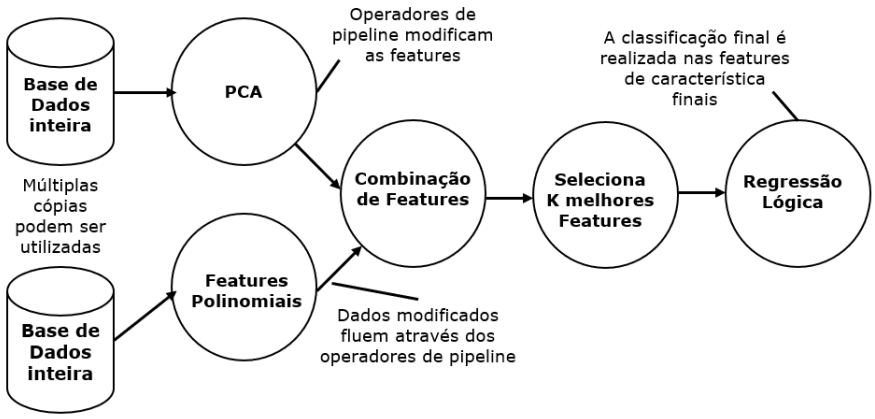
Fonte: [17].

O *Auto-Sklearn*, como mostrado na Figura 2, é divido em três partes: *meta-learning*, Otimização Bayesiana e construção de *ensembles*. O *Auto-Sklearn* se inicia com o *meta-learning* em que se inclui o conjunto de dados de entrada, logo, com base nos conjuntos de dados inicializados, já se define uma busca otimizada. A partir disso, se inicia o processo de Otimização Bayesiana, que fará paralelamente o processo de hiperparâmetros e algoritmos, aumentando a eficiência dos resultados. No final, chega então à etapa de construção de ensembles, permitindo usar todos os classificadores que foram encontrados pela Otimização Bayesiana **[2]**, selecionando os melhores algoritmos **[17]**.

2.3.3 *TPOT*

O TPOT (*Tree-based Pipeline Optimization Tool*), criado em 2016, é um sistema *AutoML* baseado em Programação Genética (*Genetic Programming*, GP) *open-source*, também baseado na API *Python Scikit-Learn* com o objetivo de maximizar a precisão da classificação em uma tarefa de classificação supervisionada. Os três operadores de *pipeline* que são utilizados no TPOT são: operador de classificação supervisionada (armazena as predições dos classificadores como uma nova *feature*), operador de pré-processamento de *feature* (modifica o conjunto de dados de alguma forma e os retornam modificados) e a seleção de *feature* (reduz o número de *features* do conjunto de dados utilizando algum critério, retornando o conjunto modificado) **[18]**. A seguir, na Figura 3, é mostrado um exemplo do *pipeline* do TPOT, cada círculo corresponde a um operador de Aprendizado de Máquina, e as setas indicam a direção dos dados.

Figura 3 – Esquema do TPOT



Fonte: [18].

3 MATERIAIS E MÉTODOS

Esta seção apresenta as análises feitas com os três sistemas de *AutoML open-source* (código aberto) mais utilizados e conhecidos pela comunidade de ciência de dados. São eles: *Auto-WEKA*, *Auto-Sklearn* e o *TPOT*.

A metodologia para o *benchmarking* foi feita da seguinte forma:

- Estudo das plataformas *open-source* existentes;
- Levantamento das características a serem comparadas;
- Agregação dos resultados.

3.1 ANÁLISE DOS ALGORITMOS SUPORTADOS

Nesta seção serão mostradas as comparações entre os três sistemas *AutoML* que podem implicar na escolha de utilizá-los.

3.1.1 Comparação dos Algoritmos de Classificação

A Tabela 1 apresenta a lista dos algoritmos de classificação suportados em cada sistema *AutoML*. É percebido que o sistema *Auto-WEKA* possui uma larga vantagem em relação ao número de algoritmos de classificação suportados.

DOI: 10.25286/repa.v7i3.2456

Tabela 1- Algoritmos de classificação suportados pelos sistemas de *AutoML* analisados

	SISTEMAS AUTOML OPEN-SOURCE		
ALGORITMOS	**Auto-WEKA**	**Auto-Sklearn**	**TPOT**
BayesNet	Sim		
DecisionStump	Sim		
DecisionTable	Sim		
GaussianProcesses	Sim		
Ibk	Sim		
Jrip	Sim		
Kstar	Sim		
LinearRegression	Sim		
LMT	Sim		
Logistic	Sim		
M5P	Sim		
M5Rules	Sim		
MultilayerPerceptron	Sim		
NaiveBayes	Sim		
NaiveBayesMultinomial	Sim		
OneR	Sim		
PART	Sim		
RandomForest	Sim	Sim	Sim
RandomTree	Sim		
REPTree	Sim		
SGD	Sim		
SimpleLinearRegression	Sim		
SimpleLogistic	Sim		
SMO	Sim		
SMOreg	Sim		
VotedPerceptron	Sim		
ZeroR	Sim		
AdaBoost (AB)		Sim	
Bernoulli naive Bayes		Sim	
Decision Tree (DT)		Sim	Sim
Extreml. Rand. Trees		Sim	
Gaussian Naive Bayes		Sim	
Gradient Boosting (GB)		Sim	
kNN		Sim	Sim
LDA		Sim	
Linear SVM		Sim	
Kernel SVM		Sim	
Multinomial Naive Bayes		Sim	
Passive Aggressive		Sim	
QDA		Sim	
Linear Class. (SGD)		Sim	
eXtreme Gradient Boosting Classifier			Sim
LogisticRegression			Sim

Fonte: Os Autores.

3.1.2 Comparação de Características dos Sistemas

A Tabela 2 apresenta características simples dos sistemas, em relação a plataforma utilizada em cada um deles e o algoritmo base da otimização que neles são feitos.

Tabela 2- Característica dos recursos de cada sistema *AutoML*

	Sistemas AutoML open-source		
Características	**Auto-WEKA**	**Auto-Sklearn**	**TPOT**
Releases	V.2.6.4	V.0.14.7	V.0.11.7
Linguagem	Java	Python	Python
Plataforma de base	WEKA	Scikit-learn	Scikit-learn
Algoritmo de previsão	Bayesiano	Bayesiano	Programação Genética
Meta-learning	Não	Sim	Sim
Ensembles	Não	Sim	Sim

Fonte: Os Autores.

3.1.3 Comparação de Cada Fase do Fluxo de Cada AutoML

A Tabela 3 faz uma comparação de cada parte do fluxo de cada sistema. São comparados: pré-processamento dos dados e engenharia de *feature*, seleção de modelos e hiperparâmetros e validação de modelo.

Tabela 3- Característica das fases de cada sistema *AutoML*

	Auto-WEKA	Auto-Sklearn	TPOT
Pré-processamento de dados e Engenharia de *features*	• A seleção de features é executada como uma fase de pré-processamento antes de construir qualquer classificador. • Um método de pesquisa combinado com um avaliador de features e os subparâmetros de ambos.	Pré-processamento de dados é feito na seguinte ordem: • One Hot Encoding nas features categóricas; • Imputação utilizando a média, mediana ou mode; • Rescaling das features; • Balanceamento do conjunto de dados utilizando os pesos das classes. Engenharia de *features*: • Matriz de decomposição usando PCA, SCV truncado, kernel PCA ou ICA; • Seleção de recursos univariada; • Seleção de *features* baseada em classificação; • Feature clustering; • Kernel approximations; • Polynomial feature expansion; • Feature embeddings; • Sparse representation and transformation.	Algoritmos suportados dos operadores de pré processamento: • StandardScaler; • RobustScaler; • MinMaxScaler; • MaxAbsScaler; • RandomizedPCA; • Binarizer; • PolynomialFeatures; Algoritmos suportados para reduzir *features*: • VarianceThreshold; • SelectKBest; • SelectPercentile; • SelectFwe; • Recursive Feature Elimination (RFE)
Seleção de modelos e hiperparâmetros	• Para a seleção são combinados 3 métodos de buscadores e 8 avaliadores, possui 2 métodos ensembles, 10 meta-métodos, 27 classificadores de base, e configurações de hiperparâmetros para cada classificador. • Otimizador Bayesiano (Bayesian Optimizer), podem ser utilizados o Sequential Modelbased Algorithm Configuration (*SMAC*) e o Tree-structured Parzen Estimator (*TPE*)	Seleção de modelos: • Resampling com otimização ensemble Otimização de hiperparâmetros: • Otimizador Bayesiano (Bayesian optimizer) + Meta learning	São suportadas implementações com os seguintes modelos de ensemble tree-based: • DecisionTree; • RandomForest; • eXtreme Gradient; • Boosting Classier (XGBoost); • LogisticRegression; • KNearestNeighborClassier; • Programação Genética (*Genetic Programming*)
Avaliação de modelos	• Cross-Validation	• Estatísticas básicas do modelo • Performance por tempo • Modelos avaliados • Tabela de liderança (*leaderboard*)	• Operador de seleção • Model dashboard

Fonte: Os Autores.

4 ANÁLISE E DISCUSSÃO DE RESULTADOS

Ao analisar cada sistema e classificar em tabelas, se pode ter uma noção de que existem correlações entre os sistemas. O *Random Forest* como algoritmo de classificação (aprendizado), é utilizado por todos os três sistemas. O *Auto-Sklearn* e o *TPOT* possuem três algoritmos em comum de aprendizado.

Não se pode deixar passar a variedade de algoritmos suportados pelo *Auto-WEKA*, há uma diferença enorme para os outros sistemas vistos. A plataforma *WEKA* se mostrou bem variada.

A linguagem suportada tanto no *Auto-Sklearn* e o *TPOT* é Python, sendo assim, mais flexível, pois já que são baseados em API *Python*, podem ser utilizados para diferentes finalidades, o *TPOT* utiliza o *meta-learning* o que pode deixar o algoritmo mais

DOI: 10.25286/repa.v7i3.2456

refinado com o resultado pretendido, do que os demais que não possuem. A linguagem *Java* utilizada por conta da plataforma *WEKA*, é um fator limitante em relação aos demais sistemas estudados.

Para a comparação mais profunda referente ao fluxo de cada sistema, temos que, o *Auto-WEKA* a seleção de *features* é executada como uma fase de pré-processamento antes de construir qualquer classificador. Para realizar a seleção de *features*, um método de pesquisa é combinado com um avaliador de *features* e os subparâmetros de ambos, sendo no máximo 5 para pesquisa e 4 para avaliação. Para a seleção são combinados 3 métodos de buscadores e 8 avaliadores, ao todo possui 2 métodos *ensembles*, 10 meta-métodos, 27 classificadores de base, e configurações de hiperparâmetros para cada classificador **[13]**. A otimização de hiperparâmetros são feitas por otimização *Bayesiana*, podendo utilizar o algoritmo SMAC ou TPE. A validação dos modelos é feita a partir de algoritmo de *cross-validation* **[19]**.

O *Auto-Sklearn* possui uma ordem de pré-processamento, respectivamente são: One Hot Encoding, imputação de dados utilizando a média, mediana ou moda, *rescaling* das *features*, balanceamento do conjunto **[20]**. Após isso e visando a otimização das *features*, as mesmas podem ser tratadas de forma opcional por um ou mais processos **[17]**. Auto-sklearn traz diversas estratégias de *resampling*, configuráveis para encontrar o melhor modelo. As estratégias de *resampling* disponíveis são: *Holdout*, *cross-validation*, *refit* e *splitter*, que são comparados pela métrica de acurácia do modelo. O *framework* é customizável para outras estratégias de *resampling* ou métricas **[21]**. Para a otimização dos modelos selecionados são aplicadas técnicas de *ensemble*, que podem ser configurados através dos hiperparâmetros: tamanho do *ensemble*, número de modelos considerados e número máximo de modelos armazenados em disco **[22]**. A otimização é feita por um otimizador *Bayesiano* combinado com *Meta Learning* **[17]**. A avaliação dos modelos é feita por estatísticas básicas, performance por tempo, e tabela de liderança **[22]**.

No sistema *TPOT* o *input* de dados é feito a partir de uma ou mais cópia dos dados, já limpos e então alimentados em um dos operadores (*preprocessing*, *decomposition*, *feature selection* e *modeling*), os dados são modificados a partir que passa por cada nó da árvore **[23]**. Pode ser implementado um operador de dimensionamento padrão que usa a média e a variância da amostra para dimensionar as features **[24]**. O *TPOT* utiliza operadores de seleção para reduzir *features* nos dados utilizando algum critério, devolvendo o conjunto de dados já modificados **[18]**. A seleção de modelos é feita com a técnica de *ensemble* na *tree-based pipeline*, após passar por todos os nós dos operadores, quando múltiplas cópias do conjunto de dados são processadas e são combinadas em um único conjunto **[23]**. A otimização dos hiperparâmetros é realizada com programação genética e a validação dos modelos é feita da seguinte forma: cada vez que o *data set* passa por um operador de seleção (por exemplo, *decision tree* ou *random forest*), os resultados das classificações são armazenados em uma coluna (chamada *guess*), de modo que o classificador mais recente para processar os dados teria as classificações nessa coluna. Após o conjunto passar completamente pelo fluxo, os valores da coluna *guess* forma utilizados para determinar a precisão e classificação **[18]**.

5 CONCLUSÕES E TRABALHOS FUTUROS

Com base nas análises obtidas anteriormente, se pode concluir que, a vasta gama de sistemas *open-source* encontrados atualmente é extremamente competitiva com os comerciais de grande porte. Sistemas de *AutoML* são de extrema importância, principalmente para quem está começando na área de ciência de dados, pois eles oferecem robustez e grande variedade de aprendizado, e além disso, uma facilidade para desenvolver soluções em análise de dados, com interfaces simples e código aberto, deixa o usuário livre para criar seu próprio roteiro de manipulação das bases de dados.

A análise dos sistemas deixa bem evidente a variedade de algoritmos possíveis do sistema *Auto-WEKA*, porém os outros sistemas não ficam para trás. O *Auto-Sklearn e o* TPOT, possuem adicionais em relação ao *Auto*-WEKA. Enquanto o sistema da plataforma *WEKA* possui um sistema mais simples de otimização Bayesiana, o *Auto-Sklearn* conta com um meta-aprendizado antes da otimização *Bayesiana*, o que já deixa a classificação mais eficiente, já que antecipou uma fase do processo. O *TPOT* possui outro tipo de otimização que se torna mais eficaz para aprendizado supervisionado, já que a interferência humana é maior.

Como trabalhos futuros, é de grande importância avaliar outros sistemas *AutoML*, não só *open-source*, mas sim os comerciais também.

REFERÊNCIAS

[1] IT CHRONICLES. **Who is Using Big Data in Business -** Disponível em: https://itchronicles.com/big-data/who-is-using-big-data-in-business/#:~:text=Despite%20the%20fact%20that%20only,at%20%2477%20billion%20by%202023.

[2] FEURER, Matthias; HUTTER, Frank. Hyperparameter optimization. In: **Automated machine learning**. Springer, Cham, p. 3-33, 2019.

[3] MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Conceitos sobre aprendizado de máquina. **Sistemas inteligentes-Fundamentos e aplicações**, v. 1, n. 1, p. 32, 2003.

[4] ERICKSON, Nick et al. Autogluon-tabular: Robust and accurate automl for structured data. **arXiv preprint arXiv:2003.06505**, 2020.

[5] FREITAS, João; LAVADO, Nuno; BERNARDINO, Jorge. Benchmarking Auto-WEKA on a Commodity Machine. In: **DATA**, p. 180-186, 2018.

[6] KRAMER, Oliver. Scikit-learn. In: **Machine learning for evolution strategies**. Springer, Cham, p. 45-53, 2016.

[7] SOUSA, Maria Cristina Cordeiro Sousa. **Uma análise do algoritmo K-means como introdução ao aprendizado de máquinas**. 2020.

[8] CLÉSIO, Flávio. Automated Machine Learning (AutoML): Aspectos práticos, teóricos, vantagens e limitações. **MEDIUM**. Disponível em: https://medium.com/data-hackers/automated-machine-learning-automl-parte-ii-2747c6237e09.

[9] GUYON, Isabelle et al. Analysis of the AutoML challenge series. **Automated Machine Learning**, p. 177, 2019.

[10] HUTTER, Frank; KOTTHOFF, Lars; VANSCHOREN, Joaquin. **Automated machine learning: methods, systems, challenges**. Springer Nature, 2019.

[11] HE, Xin; ZHAO, Kaiyong; CHU, Xiaowen. AutoML: A survey of the state-of-the-art. **Knowledge-Based Systems**, v. 212, p. 106622, 2021.

[12] CHEN, Yi-Wei; SONG, Qingquan; HU, Xia. Techniques for automated machine learning. **ACM SIGKDD Explorations Newsletter**, v. 22, n. 2, p. 35-50, 2021.

[13] THORNTON, Chris et al. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In: **Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining**, p. 847-855. 2013.

[14] THORNTON, Chris et al. Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. In: **Journal of Machine Learning Research**, p. 1-5, 2017.

[15] NAGARAJAH, Thiloshon; PORAVI, Guhanathan. A review on automated machine learning (AutoML) systems. In: **2019 IEEE 5th International Conference for Convergence in Technology (I2CT)**. p. 1-6. IEEE, 2019.

[16] ANDRADE, José. Aprendendo a aprender: O que é AutoML? **DATA SCIENCE BRIGADE**, 30, nov. 2020. Disponível em: https://blog.dsbrigade.com/introducao-a-automl/.

[17] FEURER, Matthias et al. Efficient and robust automated machine learning. **Advances in neural information processing systems**, v. 28, 2015.

[18] OLSON, Randal S.; MOORE, Jason H. TPOT: A tree-based pipeline optimization tool for automating machine learning. In: **Workshop on automatic machine learning**. PMLR, p. 66-74, 2016.

[19] MARTIN SALVADOR, Manuel; BUDKA, Marcin; GABRYS, Bogdan. Towards automatic composition of multicomponent predictive systems. In: **International conference on hybrid artificial intelligence systems**. Springer, Cham, p. 27-39, 2016.

[20] FEURER, Matthias et al. Supplementary Material for Efficient and Robust Automated Machine Learning [M]. **Advances in Neural Information Processing Systems**, 2019.

[21] AUTOML. **Resampling Strategies** – Disponível em: https://automl.github.io/auto-sklearn/development/examples/40_advanced/example_resampling.html#sphx-glr-

DOI: 10.25286/repa.v7i3.2456

examples-40-advanced-example-resampling-py.

[22] AUTOML. **Manual** – Disponível em: https://automl.github.io/auto-sklearn/development/manual.html#ensembling.

[23] OLSON, Randal S. et al. Evaluation of a tree-based pipeline optimization tool for automating data science. In: **Proceedings of the genetic and evolutionary computation conference 2016**. p. 485-492, 2016.

[24] OLSON, Randal S. et al. Automating biomedical data science through tree-based pipeline optimization. In: **European conference on the applications of evolutionary computation**. Springer, Cham, p. 123-137, 2016.