



Introdução ao AutoML

Gabriel Dantas
Eduardo de Pina
Euller Júlio
João Maurício



0 que nós abordaremos?

01

Fundamentos de AutoML

O que é e por que usar AutoML?

02

Ajuste de Modelos

Como o AutoML escolhe e ajusta o modelo automaticamente?

03

Frameworks de AutoML

Principais ferramentas que fazem AutoML.

04

Desafios e tendências

Tendências, usos práticos e limites.





01

Fundamentos de AutoML



O que é o AutoML?

AutoML (*Automated Machine Learning*) é o processo de **automatizar** as etapas envolvidas na criação de modelos de **machine learning**.

Isso inclui tarefas como **pré-processamento** de dados, **seleção de algoritmos**, **ajuste de hiperparâmetros** e até a **construção de pipelines** completos.

Objetivos do AutoML

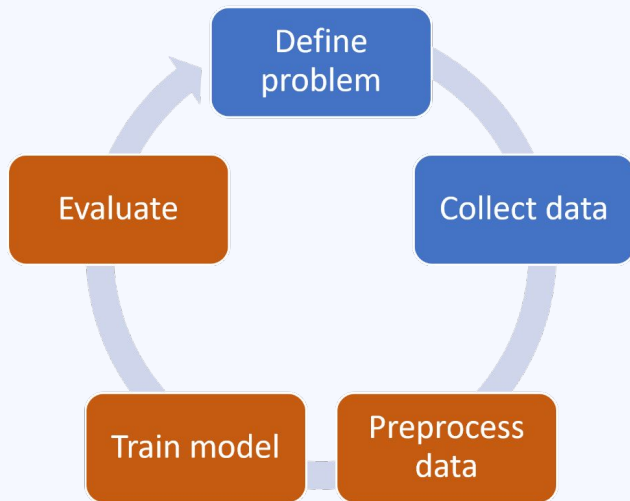
Tornar o desenvolvimento de modelos mais **acessível e eficiente**.

Isso **reduz** a dependência de especialistas, **acelerando** o ciclo de experimentação.

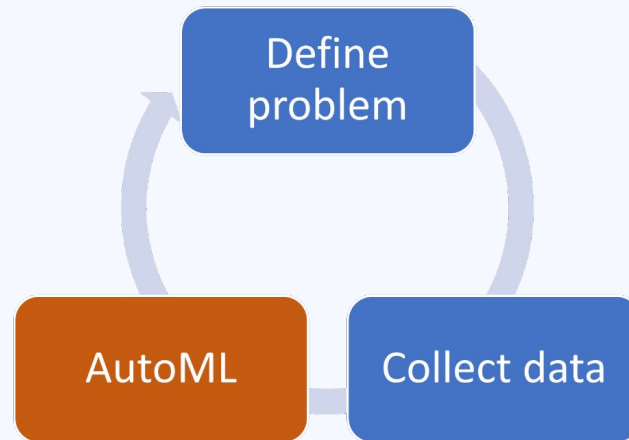
Por isso o AutoML é extremamente útil em contextos onde há **pouco conhecimento técnico**.

Ao simplificar tarefas complexas, o AutoML permite que os cientistas de dados **foquem em decisões estratégicas**, enquanto o sistema cuida do **trabalho mecânico**.

Traditional ML training workflow



AutoML workflow





Tarefas realizadas

01

Construção de Pipelines

Automatiza a escolha e a organização das etapas de pré-processamento e modelagem.

02

Escolha do modelo e otimização

Ajusta automaticamente os parâmetros dos modelos para melhorar o desempenho.

03


Neural Architecture Search

Cria e otimiza arquiteturas de redes neurais de forma inteligente.

04

Validação e avaliação automática

Executa testes e validações para garantir a robustez do modelo final.





02

Ajuste de modelos



Como o AutoML modela um problema?

01



Identificação da tarefa

Detecta se o problema é de classificação, regressão, clusterização, etc., **com base nos dados fornecidos e na variável-alvo.**

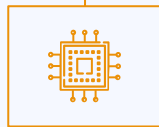
02



Análise dos dados

Verifica **tipos de variáveis**, trata **valores ausentes**, **gera estatísticas descritivas**, avalia **balanceamento** de classes etc.

03



Pré processamento automático

Codificação de variáveis categóricas, **escalonamento**, redução de **dimensionalidade**, se necessário, além da **seleção ou engenharia de features.**

04



Criação do pipeline

Monta uma **sequência de transformações e modelos** possíveis e prepara para experimentação.

05



Definição da métrica de avaliação

Ex: Acurácia, F1-score, RMSE, AUC. Pode ser **automático** ou **definido pelo usuário.**

Como o AutoML ajusta os hiperparâmetros?

01

Definição do espaço de busca

Define um espaço **contínuo** ou **discreto** de valores a serem testados. Ex: $\text{lr} \in [1e-2, 1e-6]$.

02

Seleção do método de busca

Escolhe **qual o melhor método de busca** para o problema.

• Grid Search

• Random Search

• Otimização Bayesiana

• Algoritmos evolutivos

03

Avaliação com validação cruzada

Cada combinação testada é **avaliada com cross-validation** para garantir robustez.

04

Iteração até o melhor modelo

Repete o processo até **encontrar a melhor configuração, atingir um limite de tempo ou convergência**.

Isso acelera e simplifica a implementação!

Com apenas alguns **parâmetros iniciais**, o AutoML é capaz de realizar de forma **automática** os procedimentos apresentados anteriormente.

```
clf = setup(  
    data=final_df,  
    target='phishing',  
    session_id=123,  
    train_size=0.8)
```

	Description	Value
0	Session id	123
1	Target	phishing
2	Target type	Binary
3	Original data shape	(27701, 391)
4	Transformed data shape	(27701, 391)
5	Transformed train set shape	(22160, 391)
6	Transformed test set shape	(5541, 391)
7	Numeric features	390
8	Preprocess	True
9	Imputation type	simple
10	Numeric imputation	mean
11	Categorical imputation	mode
12	Fold Generator	StratifiedKFold
13	Fold Number	10
14	CPU Jobs	-1
15	Use GPU	False
16	Log Experiment	False
17	Experiment Name	clf-default-name
18	USI	7e35

Apresentação dos melhores modelos

Após alguns minutos de execução, o AutoML retorna uma **tabela com os melhores modelos**.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lda	Linear Discriminant Analysis	0.9769	0.9963	0.9780	0.9769	0.9774	0.9539	0.9539	0.5260
ridge	Ridge Classifier	0.9755	0.9962	0.9774	0.9748	0.9761	0.9511	0.9511	0.2040
et	Extra Trees Classifier	0.9742	0.9974	0.9739	0.9756	0.9747	0.9484	0.9484	2.5540
lr	Logistic Regression	0.9735	0.9960	0.9757	0.9725	0.9741	0.9470	0.9470	1.1370
rf	Random Forest Classifier	0.9713	0.9965	0.9710	0.9728	0.9719	0.9426	0.9426	7.2930
svm	SVM - Linear Kernel	0.9639	0.9955	0.9754	0.9561	0.9652	0.9276	0.9288	0.4650
knn	K Neighbors Classifier	0.9528	0.9874	0.9789	0.9321	0.9549	0.9054	0.9066	0.8790
ada	Ada Boost Classifier	0.9313	0.9814	0.9351	0.9309	0.9329	0.8626	0.8627	10.4750
nb	Naive Bayes	0.9308	0.9831	0.9225	0.9408	0.9316	0.8615	0.8617	0.2080
qda	Quadratic Discriminant Analysis	0.8793	0.9602	0.9364	0.8946	0.9008	0.7559	0.7697	0.8470
dt	Decision Tree Classifier	0.8788	0.8788	0.8801	0.8823	0.8812	0.7575	0.7575	5.0210
dummy	Dummy Classifier	0.5109	0.5000	1.0000	0.5109	0.6763	0.0000	0.0000	0.1610



03

Frameworks de AutoML



Auto-sklearn

Origem: Universidade de Freiburg, Alemanha

Base: Scikit-learn

Diferenciais:

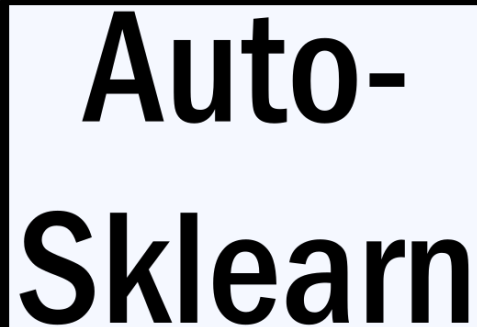
- Foco em dados tabulares
- Usa otimização Bayesiana e Meta-learning

Limitações:

- Não suporta dados de texto, imagem ou áudio
- Escalabilidade limitada para datasets grandes

Ideal para:

- Dados como planilhas e tabelas
- Aplicações de pequeno e médio porte

The logo for Auto-Sklearn, featuring the words "Auto-Sklearn" in a large, bold, black sans-serif font. The text is enclosed within a thick black rectangular border.

TPOT

Origem: Universidade de Pennsylvania.

Base: Scikit-learn + Programação genética

Diferenciais:

- Usa algoritmos genéticos para evoluir pipelines
- Gera código python automatizado automaticamente

Limitações:

- Tempo de execução elevado em datasets grandes
- Sem suporte nativo para dados não tabulares

Ideal para:

- Prototipagem rápida
- Geração de pipelines replicáveis e mais compreensíveis



PyCaret

Origem: Comunidade open-source.

Base: Scikit-learn, XGBoost, LightGBM, CatBoost.

Diferenciais:

- Implementação extremamente simples e intuitiva
- Facilita o deploy e integração com streamlit, flask e cloud

Limitações:

- Performance inferior em problemas muito complexos
- Dependências de bibliotecas externas para tarefas muito específicas

Ideal para:

- Pequenas e médias empresas
- Automação de tarefas comuns em análise de dados

The PyCaret logo features the word "PYCARET" in a bold, sans-serif font. The "PY" is in a light blue color, while "CARET" is in black. The letters are closely spaced, and the overall design is modern and professional.

FLAML

Origem: Microsoft Research.

Base: algoritmo próprio (BlendSearch)

Diferenciais:

- Foco em baixo custo computacional
- Extremamente rápido e eficiente em datasets grandes

Limitações:

- Suporte limitado para deep learning
- Não realiza etapas avançadas de pré-processamento de dados

Ideal para:

- Ambientes com recursos limitados
- Aplicações rápidas e prototipagem leve





04

Desafios e tendências



The top left corner features a decorative graphic of circuit lines. It includes a grid of small blue dots, with orange and purple lines weaving through them. Several circular nodes, some orange and some purple, are placed at various points along these lines.

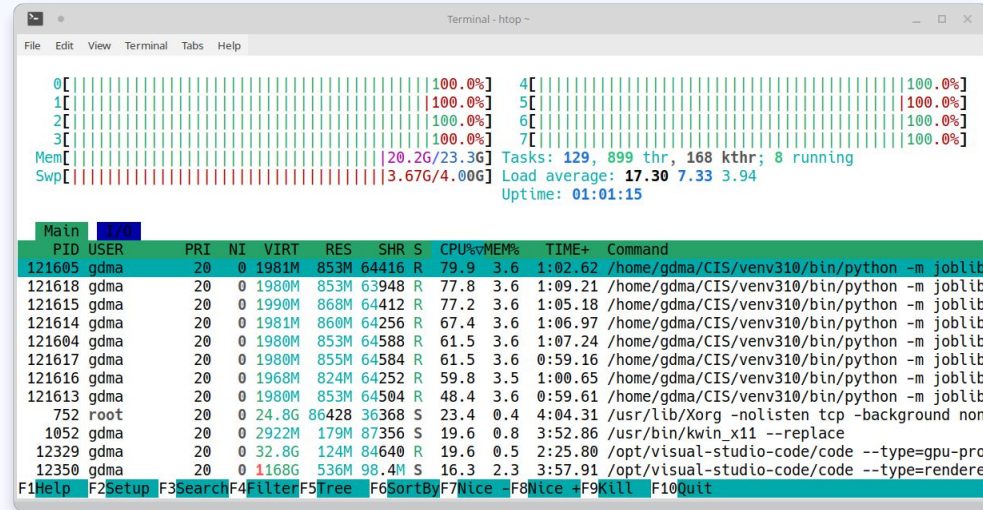
Nem tudo são flores...



Alto custo computacional

O AutoML pode **consumir muitos recursos** para explorar o espaço de modelos e hiperparâmetros.

Isso **aumenta o custo computacional** e muitas vezes o **tempo de execução**.



The screenshot shows a terminal window titled "Terminal - htop". The top section displays system statistics: CPU usage at 100.0% across 8 cores, memory usage at 20.26/23.36 GB, and swap usage at 3.67G/4.00G. It also shows 129 tasks, 899 threads, and 168 kthrs. The bottom section is a table of running processes.

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
121605	gdma	20	0	1981M	853M	64416	R	79.9	3.6	1:02.62	/home/gdma/CIS/venv310/bin/python -m joblib
121618	gdma	20	0	1980M	853M	63948	R	77.8	3.6	1:09.21	/home/gdma/CIS/venv310/bin/python -m joblib
121615	gdma	20	0	1990M	868M	64412	R	77.2	3.6	1:05.18	/home/gdma/CIS/venv310/bin/python -m joblib
121614	gdma	20	0	1981M	860M	64256	R	67.4	3.6	1:06.97	/home/gdma/CIS/venv310/bin/python -m joblib
121604	gdma	20	0	1980M	853M	64588	R	61.5	3.6	1:07.24	/home/gdma/CIS/venv310/bin/python -m joblib
121617	gdma	20	0	1980M	855M	64584	R	61.5	3.6	0:59.16	/home/gdma/CIS/venv310/bin/python -m joblib
121616	gdma	20	0	1968M	824M	64252	R	59.8	3.5	1:00.65	/home/gdma/CIS/venv310/bin/python -m joblib
121613	gdma	20	0	1980M	853M	64504	R	48.4	3.6	0:59.61	/home/gdma/CIS/venv310/bin/python -m joblib
752	root	20	0	24.8G	86428	36368	S	23.4	0.4	4:04.31	/usr/lib/Xorg -nolisten tcp -background non
1052	gdma	20	0	2922M	179M	87356	S	19.6	0.8	3:52.86	/usr/bin/kwin_x11 --replace
12329	gdma	20	0	32.8G	124M	84640	R	19.6	0.5	2:25.80	/opt/visual-studio-code/code --type=gpu-pro
12350	gdma	20	0	1168G	536M	98.4M	S	16.3	2.3	3:57.91	/opt/visual-studio-code/code --type=rende


At the bottom, there is a navigation bar with keys: F1Help, F2Setup, F3Search, F4Filter, F5Tree, F6SortBy, F7Nice, F8Nice, F9Kill, F10Quit.

Pouca interpretabilidade

Ao focar em **maximizar a performance**, o AutoML frequentemente gera modelos que **tendem a ser mais complexos e difíceis de explicar**.


Em **áreas delicadas**, como **segurança e saúde**, é fundamental compreender e justificar as decisões do modelo. Isso se torna um **desafio com AutoML**.

Por isso o AutoML é utilizado preferencialmente como **uma ferramenta de prototipagem** e de **“Proof Of Concept”**.



“It’s **very difficult** to understand the **process** and the **outcomes** from those techniques. It’s also difficult to figure out **whether we can trust the models and whether we can make fair decisions when using them.**”

**Interpretability is crucial for trusting
AI and machine learning - SAS**



Limitações para problemas específicos

Pode **não se adaptar bem a tarefas muito customizadas e específicas.**

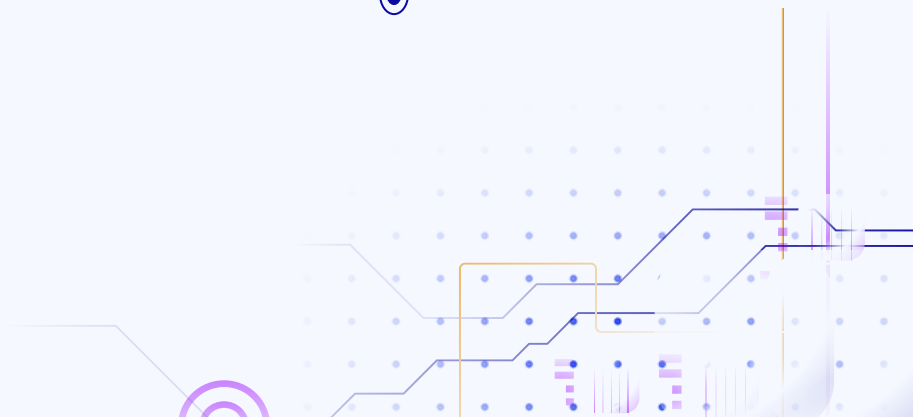
(**Ex:** interpretabilidade legal, tempo real).

Sistemas que exigem **respostas em tempo real ou latência baixa** geralmente **não alcançam o melhor desempenho** com AutoML, devido à **complexidade** dos modelos gerados.

Essa mesma complexidade também limita a implementação de AutoML em **sistemas embarcados.**

The top left corner features a decorative graphic of circuit lines in orange and purple, with small circles at the junctions, set against a light blue background with a grid of dots.

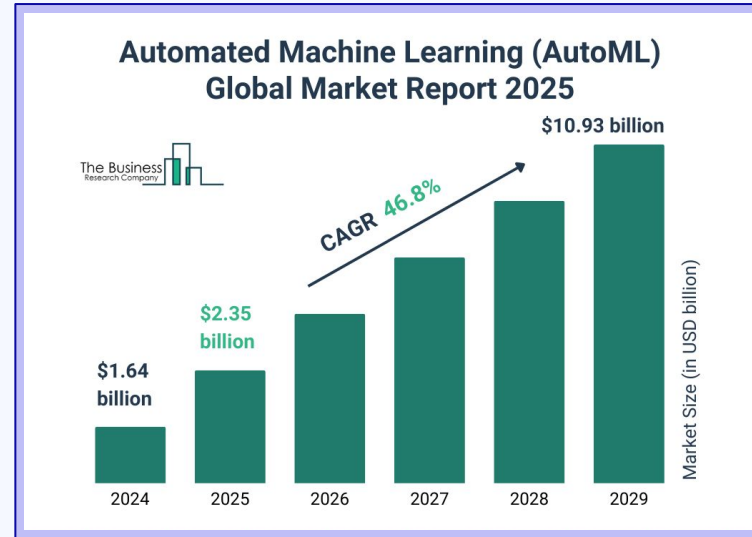
E quais as tendências futuras?



Democratização e integração

Expansão do uso em negócios de todos os portes, inclusive em pequenas empresas.

Isso consequentemente **amplia o acesso à inteligência artificial**, estimulando a **inovação** e sua adoção em áreas não tradicionais.

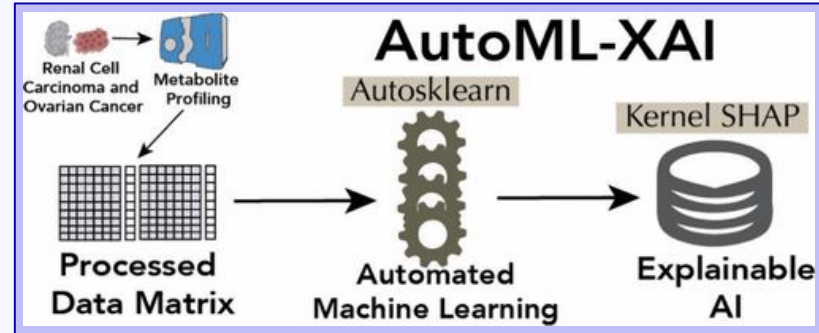


AutoML com explicabilidade

Cresce a demanda por **modelos mais interpretáveis**.

Isso impulsiona a combinação de **AutoML com XAI** (Explainable AI).

O objetivo é manter a automação no desenvolvimento de modelos, **sem abrir mão da transparência e clareza** das decisões geradas.



Artigo: Automated Machine Learning and Explainable AI (AutoML-XAI) for Metabolomics: Improving Cancer Diagnostics

AutoML Sustentável e Eficiente

Otimização do consumo energético dos processos de treinamento e busca de hiperparâmetros, alinhando-se às práticas de **Green AI**.

A tendência é tornar o AutoML mais **leve, rápido e energeticamente eficiente**.

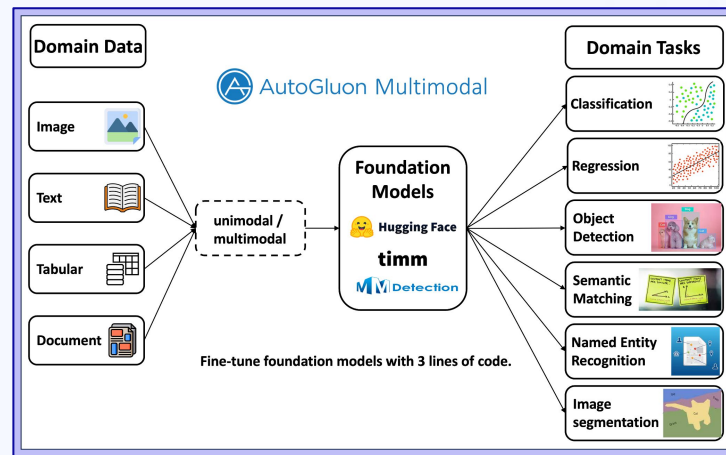


www.automl.org/green-automl

Implementação de AutoML multimodal

Integração de dados de diferentes naturezas – texto, imagem, documentos, tabelas – **em um único pipeline automatizado.**

Isso permitirá a existência de modelos **mais robustos e capazes de lidar com diferentes tipos de informação.**



auto.gluon.ai



Obrigado !