



AUTOMAÇÃO DE MACHINE LEARNING

- AutoML: é o uso de ferramentas que automatizam tarefas do pipeline de machine learning, como pré-processamento, seleção de modelo e ajuste de hiperparâmetros.
- Tem por objetivo democratizar o acesso à IA e acelerar o desenvolvimento de modelos preditivos.

DESAFIOS DO ML TRADICIONAL

- **Pipeline complexo:** envolve várias etapas manuais.
- **Exige conhecimento avançado:** estatística, programação, engenharia de dados.
- **Tempo e recursos:** muitos testes e ajustes consomem tempo.

BENEFÍCIOS DO AUTOML

- **Produtividade:** reduz o tempo de desenvolvimento.
- **Precisão:** escolhe os melhores modelos e hiperparâmetros.
- **Acessibilidade:** permite que não-especialistas utilizem ML com qualidade.

PYCARET

- Biblioteca de AutoML em Python com foco em simplicidade e eficiência.
- Baseado no Scikit-learn, mas com comandos mais compactos e intuitivos.

PRINCIPAIS FUNCIONALIDADES

- `setup()`: Inicializa o ambiente.
- `compare_models()`: Compara dezenas de modelos.
- `tune_model()`: Ajusta hiperparâmetros automaticamente.
- `predict_model()`: Gera previsões.
- `save_model()`, `deploy_model()`: Exporta e implanta modelos.

VANTAGENS DO PYCARET

- **Facilidade de uso:** Interface de alto nível.
- **Integração:** Funciona bem com pandas, Jupyter, MLFlow.
- **Diversidade:** suporta classificação, regressão, clustering, NLP, etc.

PIPELINE TÍPICO DO PYCARET

- Carregamento de dados.
- `setup()` para preparar o ambiente.
- `compare_models()` para avaliar.
- `tune_model()` para otimizar.
- `finalize_model()` para consolidar.

INTERPRETAÇÃO DOS RESULTADOS

- Tabela com métricas: Accuracy, ROC AUC, Recall, F1.
- Visualizações: Matriz de confusão, curva ROC.

TUNING E ENSEMBLE

- **tune_model()**: Busca hiperparâmetros ideais.
- **blend_models()** e **stack_models()**: Combina modelos para melhor desempenho.

INTERPRETAÇÃO COM SHAP

- **interpret_model()**: Explica predições com SHAP.
- Importância de variáveis, impactos locais/globais.

EXPORTAÇÃO E IMPLANTAÇÃO

- **save_model()** e **load_model()** para salvar e reutilizar.
- **deploy_model()** para produção (ex: AWS, GCP, Flask).
- MLOps

FERRAMENTAS DE AUTOML

Ferramenta	Exemplo de uso	Linguagem	Explicabilidade
PyCaret	Classificação, regressão, clustering, séries temporais	Python	Sim (SHAP, gráficos nativos)
H2O AutoML	Classificação e regressão em escala corporativa	Python, R, Java	Sim (SHAP, LIME)
MLJAR-supervised	Geração de modelos com relatórios HTML automáticos	Python	Sim (visualizações automáticas)
Auto-sklearn	Pesquisa acadêmica e competição de ML	Python	Não nativamente
TPOT	Otimização de pipelines com algoritmos genéticos	Python	Não nativamente
Google Cloud AutoML	Classificação e regressão com interface web	Interface gráfica, Python	Sim (parcial)
Amazon SageMaker Autopilot	Modelos em escala na AWS	Interface gráfica, Python	Sim (com SageMaker Clarify)



UNITINS

UNIVERSIDADE ESTADUAL DO TOCANTINS