MACHINE LEARNING ACESSÍVEL: DESENVOLVIMENTO DE UMA APLICAÇÃO INTERATIVA COM STREAMLIT E PYCARET

Raffael Guideti Miello

Acadêmico do Curso de Engenharia de Software do Centro Universitário Cesumar – UNICESUMAR, Curitiba – PR. Bolsista PIBIC/CNPq-UniCesumar. raffaelguideti@gmail.com

**RESUMO:** Este artigo apresenta uma aplicação interativa de Machine Learning desenvolvida com Streamlit e PyCaret, visando simplificar o processo de análise e modelagem de dados. A metodologia envolve a integração dessas duas ferramentas para criar uma plataforma acessível que permite o upload de dados, análise exploratória, seleção de variáveis, treinamento e comparação de modelos (classificação, regressão e clusterização), análise de desempenho e previsão com novos dados. Os resultados demonstram a eficácia da combinação Streamlit-PyCaret na democratização do Machine Learning, oferecendo uma solução ágil e intuitiva para usuários com diferentes níveis de conhecimento técnico. A aplicação abstrai a complexidade subjacente da modelagem, permitindo foco na interpretação e aplicação dos resultados.

**PALAVRAS-CHAVE:** Machine Learning; Streamlit; PyCaret; Análise de Dados; Modelagem Preditiva.

**INTRODUÇÃO**

 A crescente demanda por soluções baseadas em dados tem impulsionado o desenvolvimento de ferramentas que tornam o Machine Learning (ML) mais acessível e eficiente. Tradicionalmente, a construção de modelos de ML envolve um processo complexo que abrange desde a coleta e pré-processamento de dados até o treinamento, avaliação e implantação dos modelos. Essa complexidade, muitas vezes, exige um conhecimento aprofundado em programação, estatística e algoritmos, criando uma barreira significativa para profissionais de diversas áreas que poderiam se beneficiar do uso de ML.

Neste contexto, a busca por plataformas que simplifiquem o ciclo de vida do ML, sem comprometer a flexibilidade e o poder analítico, tornou-se uma prioridade. Ferramentas low-code e frameworks de desenvolvimento rápido de aplicações web surgem como soluções promissoras para preencher essa lacuna. O Streamlit, por exemplo, permite a criação de interfaces de usuário interativas com poucas linhas de código Python, enquanto o PyCaret automatiza grande parte do fluxo de trabalho de ML, desde a preparação dos dados até a seleção e otimização de modelos.

O presente trabalho propõe o desenvolvimento de uma aplicação interativa de Machine Learning que integra o Streamlit e o PyCaret. O objetivo principal é oferecer uma plataforma intuitiva e de fácil utilização para que usuários possam realizar análises exploratórias de dados, treinar e comparar diferentes modelos de ML (classificação, regressão e clusterização), analisar o desempenho desses modelos e realizar previsões com novos conjuntos de dados. A aplicação visa, portanto, democratizar o acesso às capacidades do Machine Learning, permitindo que profissionais de diversas áreas possam explorar e aplicar modelos preditivos sem a necessidade de um conhecimento aprofundado em codificação. Este artigo detalha a arquitetura da solução, as funcionalidades implementadas e os benefícios dessa abordagem integrada.

**1 MÉTODOS DE ANÁLISE:**

**1.1 Arquitetura da solução:**

Os principais componentes da arquitetura incluem:

• Streamlit (Frontend): Responsável pela interface gráfica do usuário, permitindo a interação com a aplicação através de upload de arquivos, seleção de opções e visualização de resultados. É a camada visível ao usuário, construída para ser intuitiva e responsiva.

• PyCaret (Backend): Biblioteca de Machine Learning low-code que abstrai o processo de modelagem, incluindo pré-processamento de dados, treinamento, avaliação e comparação de modelos. O PyCaret gerencia a complexidade dos algoritmos de ML, permitindo que o Streamlit se concentre na apresentação.

• Pandas: Utilizado para manipulação e análise de dados em memória. É a base para o tratamento dos dados carregados e para a interação entre o Streamlit e o PyCaret.

• Docker (Opcional): Ferramenta de conteinerização que pode ser utilizada para empacotar a aplicação e suas dependências, garantindo portabilidade e reprodutibilidade em diferentes ambientes de execução.

• Banco de Dados SQL (Opcional/Futuro): Previsão para integração futura para persistência de dados e modelos, o que permitiria escalabilidade e gerenciamento eficiente de grandes volumes de informação, além de um controle de versão de modelos.

**1.2 Funcionalidades da aplicação:**

A aplicação oferece um conjunto abrangente de funcionalidades que cobrem as etapas essenciais de um projeto de Machine Learning:

• Upload de Dados: A interface permite que o usuário carregue seus próprios datasets nos formatos CSV ou Excel. Os dados são automaticamente lidos e preparados para as etapas subsequentes.

• Análise Exploratória de Dados (EDA): Após o upload, a aplicação exibe estatísticas descritivas básicas, tipos de dados e a contagem de valores ausentes para cada coluna. Isso fornece um panorama inicial da qualidade e distribuição dos dados.

• Seleção de Variáveis: O usuário pode selecionar a variável alvo (dependente) para problemas de classificação e regressão, indicando qual coluna será utilizada para o treinamento do modelo.

• Seleção do Tipo de Problema: A aplicação é flexível e permite que o usuário escolha o tipo de problema de Machine Learning a ser resolvido: Classificação, Regressão ou Clusterização. Essa escolha adapta as opções de modelagem e as métricas de avaliação.

• Treinamento e Comparação de Modelos (PyCaret): Com base no tipo de problema selecionado, o PyCaret é invocado para configurar o ambiente de ML, treinar e comparar automaticamente diversos algoritmos. Os resultados da comparação, incluindo métricas de desempenho, são apresentados de forma clara.

• Análise do Modelo: Para o melhor modelo identificado, a aplicação permite a visualização de gráficos específicos que auxiliam na análise de desempenho, como Curvas ROC, Matrizes de Confusão (para classificação), Gráficos de Resíduos (para regressão) e Gráficos de Elbow/Silhueta (para clusterização).

• Previsão com Novos Dados: Uma funcionalidade crucial é a capacidade de carregar novos dados e utilizar o modelo treinado para gerar previsões. Isso simula um cenário de implantação, onde o modelo é aplicado a dados não vistos.

**1.3 Fluxo de Interação:**

O fluxo de interação do usuário com a aplicação é intuitivo e segue uma sequência lógica:

1. Início: O usuário acessa a aplicação Streamlit via navegador web.

2. Carregamento de Dados: Um arquivo CSV ou Excel é carregado através da interface. O Pandas é utilizado para processar o arquivo e criar um DataFrame.

3. Exploração Inicial: O usuário visualiza as estatísticas descritivas e informações sobre os dados para uma compreensão inicial.

4. Definição do Problema: A variável alvo é selecionada (para classificação/regressão) e o tipo de problema de ML é definido.

5. Execução do PyCaret: Ao acionar o botão “Executar PyCaret”, o ambiente do PyCaret é configurado com os dados e o tipo de problema. Em seguida, os modelos são treinados e comparados.

6. Avaliação e Análise: Os resultados da comparação de modelos são exibidos, e o usuário pode gerar gráficos para uma análise mais aprofundada do modelo selecionado.

7. Persistência e Previsão: O melhor modelo é salvo em disco. Posteriormente, novos dados podem ser carregados para que o modelo salvo realize previsões.

Este fluxo simplifica o processo de Machine Learning, tornando-o acessível mesmo para usuários sem experiência prévia em codificação de ML.

**CONCLUSÃO**

A aplicação interativa de Machine Learning desenvolvida, integrando Streamlit e PyCaret, demonstra um avanço significativo na democratização do acesso a ferramentas de análise e modelagem de dados. Ao abstrair a complexidade inerente ao ciclo de vida do Machine Learning, a plataforma permite que usuários com diferentes níveis de conhecimento técnico possam realizar análises exploratórias, treinar e comparar modelos preditivos, e gerar previsões de forma intuitiva e eficiente. Os resultados obtidos com a integração dessas tecnologias ressaltam a agilidade no desenvolvimento e a capacidade de entregar soluções robustas para problemas de classificação, regressão e clusterização.

Um dos principais aprendizados deste projeto foi a validação da abordagem low-code para o desenvolvimento de aplicações de ML. A combinação de Streamlit, para a construção rápida de interfaces de usuário, e PyCaret, para a automação do fluxo de trabalho de ML, provou ser extremamente eficaz na redução do tempo e esforço necessários para prototipar e implantar modelos. Além disso, a modularidade da arquitetura facilita a manutenção e a inclusão de novas funcionalidades.

**REFERÊNCIAS**

STREAMLIT. Streamlit. Disponível em: https://streamlit.io/. Acesso em: 16 jun. 2025.

PYCARET. PyCaret. Disponível em: https://pycaret.org/. Acesso em: 16 jun. 2025.

PANDAS. Pandas. Disponível em: https://pandas.pydata.org/. Acesso em: 16 jun. 2025.

DOCKER. Docker. Disponível em: https://www.docker.com/. Acesso em: 16 jun. 2025.

KAGGLE. Kaggle. Disponível em: https://www.kaggle.com/. Acesso em: 16 jun. 2025.

YDATA. ydata-profiling. Disponível em: https://ydata-profiling.ydata.ai/. Acesso em: 16 jun. 2025.

D-TALE. D-Tale documentation. Disponível em: https://dtale.readthedocs.io/en/latest/. Acesso em: 16 jun. 2025.

**ACCESSIBLE MACHINE LEARNING: DEVELOPMENT OF AN INTERACTIVE APPLICATION WITH STREAMLIT AND PYCARET**

**ABSTRACT:** In this article, we present an interactive Machine Learning application developed with Streamlit and PyCaret, aiming to simplify the data analysis and modeling process. The methodology involves integrating these two tools to create an accessible platform that allows data upload, exploratory analysis, variable selection, training and comparison of models (classification, regression, and clustering), performance analysis, and prediction with new data. The results demonstrate the effectiveness of the Streamlit-PyCaret combination in democratizing Machine Learning, offering an agile and intuitive solution for users with varying levels of technical knowledge. The application abstracts the underlying complexity of modeling, allowing focus on the interpretation and application of results.

**KEYWORDS:** Machine Learning; Streamlit; PyCaret; Data Analysis; Predictive Modeling.