## **Sistema Inteligente de Análise e Modelagem de Dados com PyCaret**

Matheus Mitsuo Yamafuku Benatti

### **Visão Geral**

O presente trabalho descreve o desenvolvimento de uma aplicação web interativa denominada ML Data HUB, voltada à experimentação e operacionalização de projetos de ciência de dados com foco em aprendizado de máquina. A aplicação foi construída em Python, utilizando o framework Streamlit como camada de interface, e adota como núcleo de backend a biblioteca PyCaret, com o objetivo de oferecer uma solução robusta, modular e acessível para todo o ciclo de vida de projetos de modelagem preditiva e exploratória.

O sistema foi concebido para atender a diversos perfis de usuários, desde iniciantes em ciência de dados até analistas experientes, promovendo a integração fluida de etapas fundamentais da análise de dados. Dentre os principais recursos contemplados, destacam-se:

1. o upload de qualquer base de dados tabular no formato CSV ou Excel;
2. a realização de análise exploratória de dados (EDA) com suporte a múltiplas bibliotecas de visualização e estatística;
3. a seleção manual de variáveis independentes e dependentes para os modelos;
4. a escolha entre três paradigmas de aprendizado de máquina: classificação, regressão e clusterização, com visualização integrada dos resultados;
5. o treinamento e comparação automatizada de modelos por meio do PyCaret, incluindo análise de desempenho e tuning de hiperparâmetros;
6. a persistência de dados e modelos treinados em um banco de dados relacional via SQLAlchemy, com possibilidade de versionamento;
7. a realização de previsões com novos dados inseridos dinamicamente, simulando ambientes reais de aplicação.

A biblioteca PyCaret é responsável por simplificar operações tradicionalmente complexas em machine learning, como o pré-processamento, avaliação cruzada, ordenação por métricas, e exportação dos pipelines. Essa abordagem está alinhada com as tendências contemporâneas de ferramentas low-code/no-code, que visam democratizar o uso da inteligência artificial (Ali et al., 2020).

Complementarmente, a aplicação utiliza as bibliotecas pandas (McKinney, 2010), scikit-learn (Pedregosa et al., 2011), matplotlib e seaborn para permitir uma manipulação eficiente de dados e uma visualização analítica informativa, como preconizado por Tukey (1977) na análise exploratória.

O projeto adota uma arquitetura modular baseada em boas práticas de engenharia de software, com divisão clara entre as camadas de entrada, tratamento, modelagem e persistência. A gestão de estado via variáveis de sessão, a separação de responsabilidades em módulos específicos (por exemplo, data\_processing, model\_selection, predictions), e o uso de banco de dados relacional garantem reprodutibilidade, rastreabilidade e escalabilidade do sistema.

Por fim, o design intuitivo da interface, aliado à automatização inteligente de tarefas técnicas, permite que o ML Data HUB funcione como uma ferramenta pedagógica e operacional, contribuindo tanto para o ensino quanto para a aplicação prática de ciência de dados em diferentes domínios.

**Fundamentos Teóricos**

A concepção e implementação do sistema ML Data HUB fundamenta-se em princípios consolidados da ciência de dados, aprendizado de máquina automatizado (AutoML) e engenharia de software aplicada à análise de dados. A arquitetura modular da aplicação reflete as melhores práticas em cada uma das etapas do pipeline de machine learning, desde a ingestão dos dados até a geração de predições a partir de modelos treinados. Os principais pilares teóricos que embasam o sistema são descritos a seguir:

**1. Análise Exploratória de Dados (EDA)**

A Análise Exploratória de Dados é uma etapa crítica no processo de descoberta de conhecimento em bases de dados (KDD), tendo sido consolidada por John Tukey (1977) como uma abordagem essencial para a formulação de hipóteses, detecção de padrões e compreensão das estruturas subjacentes nos dados. No ML Data HUB, a EDA é implementada por meio de ferramentas estatísticas e gráficas que incluem:

* Cálculo de medidas descritivas (média, mediana, desvio padrão);
* Visualização de distribuições e boxplots para avaliação de assimetria e outliers;
* Mapas de calor para análise de correlação;
* Avaliação de valores ausentes e cardinalidade de atributos.

Essas funcionalidades são viabilizadas pelo uso de bibliotecas como pandas (McKinney, 2010), matplotlib, seaborn (Waskom, 2021) e, em versões mais completas, ydata-profiling e Sweetviz, que facilitam uma compreensão visual abrangente dos dados e promovem diagnósticos fundamentados para as etapas subsequentes.

**2. Pré-processamento de Dados**

A qualidade dos dados influencia diretamente a eficácia dos modelos de machine learning. Dessa forma, o pré-processamento representa uma etapa indispensável para assegurar a integridade, consistência e adequação estatística dos dados. As técnicas aplicadas no ML Data HUB incluem:

* Normalização de colunas e padronização de nomes;
* Tratamento de valores ausentes (remoção ou imputação estatística/heurística);
* Redução de cardinalidade para variáveis categóricas (frequência, binning);
* Remoção de outliers com base no método interquartílico (IQR);
* Criação de variáveis derivadas, como a extração de idade a partir de datas;
* Codificação de variáveis categóricas, incluindo técnicas como One-Hot e Target Encoding.

Esses procedimentos seguem diretrizes consagradas na literatura (Kuhn & Johnson, 2019; Géron, 2019) e são executados de forma semiautomática, preparando os dados de maneira robusta para a etapa de modelagem.

**3. Modelagem Automatizada com PyCaret**

A modelagem preditiva é conduzida com auxílio da biblioteca PyCaret (Ali et al., 2020), que automatiza processos complexos de machine learning sem sacrificar a flexibilidade analítica. O sistema suporta três paradigmas fundamentais:

* Classificação, para problemas com variáveis-alvo discretas;
* Regressão, para variáveis contínuas;
* Clusterização, para agrupamento não supervisionado.

O PyCaret permite:

* Comparação de dezenas de algoritmos com validação cruzada;
* Ajuste automático de hiperparâmetros;
* Criação de pipelines reutilizáveis e exportáveis;
* Geração de métricas padronizadas (ROC AUC, R², silhouette score, entre outras);
* Interpretação de modelos com SHAP, LIME e gráficos de importância.

A biblioteca é compatível com algoritmos de ponta como Random Forest, XGBoost, CatBoost, LightGBM e KMeans, e utiliza como backend o scikit-learn (Pedregosa et al., 2011), assegurando reprodutibilidade e rastreabilidade estatística dos modelos treinados.

**4. Interface Interativa com Streamlit**

A camada de apresentação do sistema foi desenvolvida utilizando Streamlit (2024), um framework moderno para construção de interfaces gráficas voltadas à ciência de dados. A escolha por Streamlit está alinhada com a proposta de democratização do acesso ao aprendizado de máquina, pois elimina a necessidade de programação direta por parte do usuário.

A interface permite:

* Upload e visualização de datasets;
* Seleção manual de atributos preditores e variável-alvo;
* Escolha do tipo de modelo desejado (classificação, regressão, clusterização);
* Visualização de gráficos interativos e métricas de desempenho;
* Inserção de novos dados para previsão em tempo real.

Essa abordagem facilita o uso da aplicação em contextos educacionais, institucionais ou corporativos, promovendo acessibilidade, transparência e engajamento.

**5. Persistência, Versionamento e Gestão de Resultados**

A persistência dos dados, modelos e metainformações é gerida por meio do SQLAlchemy (Bayer, 2024), que oferece uma abstração poderosa sobre bancos de dados relacionais. A adoção dessa abordagem traz benefícios relevantes:

* Armazenamento sistemático de experimentos e predições;
* Versionamento de modelos e datasets;
* Reuso de pipelines e resultados em execuções futuras;
* Geração de logs para auditoria e rastreabilidade.

Essas práticas estão em consonância com o movimento de MLOps (Machine Learning Operations), que busca integrar princípios de engenharia de software às rotinas de ciência de dados, como sugerido por Sculley et al. (2015).

**Gráficos e relatórios**

No contexto do sistema ML Data HUB, a geração de gráficos e relatórios desempenha papel central na análise, interpretação e comunicação dos resultados obtidos ao longo do pipeline de ciência de dados. A visualização de dados, fundamentada em princípios estatísticos e de design informacional, é reconhecida como uma das etapas mais importantes para a compreensão de padrões, tendências e anomalias em conjuntos de dados complexos (Tukey, 1977; Waskom, 2021).

Durante a análise exploratória de dados (EDA), o sistema disponibiliza uma variedade de gráficos, tais como histogramas, gráficos de densidade, gráficos de barras para variáveis categóricas, e mapas de calor para matrizes de correlação. Essas visualizações são implementadas com o auxílio das bibliotecas Matplotlib e Seaborn, que oferecem recursos robustos para a construção de gráficos estatísticos em Python (Hunter, 2007; Waskom, 2021). A escolha adequada do tipo de gráfico para cada variável ou relação investigada facilita a identificação de distribuições, outliers, relações lineares ou não lineares e agrupamentos naturais nos dados (Géron, 2019).

Após o treinamento dos modelos de aprendizado de máquina, o sistema gera relatórios de desempenho que incluem tanto métricas quantitativas (como acurácia, precisão, recall, F1-score, R², RMSE, silhouette score) quanto gráficos específicos para cada tarefa. Por exemplo, para classificação, são apresentados a matriz de confusão e a curva ROC; para regressão, gráficos de resíduos e comparação entre valores reais e previstos; para clusterização, gráficos de silhueta e visualização dos clusters. Esses relatórios são essenciais para a avaliação crítica dos modelos, permitindo ao usuário comparar diferentes algoritmos e identificar possíveis limitações ou oportunidades de melhoria (Pedregosa et al., 2011; Ali et al., 2020).

A integração dos gráficos e relatórios à interface interativa, por meio do Streamlit, potencializa a experiência do usuário, tornando o processo de análise mais dinâmico, transparente e acessível. Além disso, a apresentação visual dos resultados contribui para a comunicação efetiva dos achados, tanto para públicos técnicos quanto não técnicos, aspecto fundamental em projetos de ciência de dados aplicados (Knaflic, 2015).

**Funcionalidades**

O sistema ML Data HUB oferece um conjunto abrangente de funcionalidades voltadas ao suporte completo do ciclo de vida de projetos de ciência de dados e machine learning. A seguir, são detalhados os principais componentes e suas respectivas contribuições metodológicas e técnicas:

**1. Carregamento e Armazenamento de Dados**

O sistema disponibiliza um módulo robusto para importação de bases de dados, compatível com os formatos .csv e .xlsx, permitindo ao usuário carregar arquivos locais por meio de uma interface intuitiva desenvolvida em Streamlit. Durante o processo de ingestão, são realizadas rotinas automáticas de:

* Detecção e padronização de nomes de colunas;
* Conversão de tipos de dados;
* Tratamento inicial de valores ausentes;
* Identificação de variáveis categóricas e numéricas.

Uma vez carregados, os dados podem ser armazenados em um banco de dados relacional utilizando SQLAlchemy, com persistência de metadados como:

* Nome do dataset;
* Número de registros e atributos;
* Tipo de tarefa associada (classificação, regressão ou clusterização);
* Data e hora da inserção.

Essa funcionalidade assegura a rastreabilidade e o versionamento dos dados, aspectos fundamentais no contexto de experimentos reprodutíveis e auditáveis (Sculley et al., 2015).

**2. Análise Exploratória de Dados (EDA)**

A aplicação disponibiliza ferramentas interativas e automatizadas para a análise exploratória dos dados, permitindo ao usuário:

* Visualizar estatísticas descritivas por variável;
* Detectar outliers por meio de boxplots;
* Analisar distribuições com histogramas, gráficos de densidade e barras;
* Avaliar correlações entre variáveis com mapas de calor;
* Observar a presença e localização de valores ausentes.

A EDA é fundamental para revelar estruturas latentes, comportamentos atípicos e relações entre variáveis, influenciando diretamente decisões de pré-processamento e modelagem (Tukey, 1977). Além disso, a interface facilita a leitura desses resultados por meio de visualizações geradas com matplotlib, seaborn e bibliotecas complementares, conforme a configuração do sistema.

**3. Seleção de Variáveis e Definição da Tarefa de Modelagem**

O sistema fornece controle total ao usuário sobre a especificação do problema de aprendizado de máquina. Entre as opções disponíveis estão:

* Seleção da variável alvo (target);
* Escolha das variáveis preditoras a serem utilizadas;
* Definição da tarefa de modelagem:
  + Classificação para alvos categóricos;
  + Regressão para variáveis contínuas;
  + Clusterização para agrupamentos não supervisionados.

Para tarefas de clusterização, o sistema permite configurar o algoritmo (por exemplo, KMeans, Birch ou DBSCAN) e o número de clusters. Essa etapa é essencial para garantir que o modelo escolhido esteja alinhado com a natureza estatística do problema em questão (Kuhn & Johnson, 2019).

**4. Treinamento, Comparação e Salvamento de Modelos**

Com o uso da biblioteca PyCaret, o sistema automatiza todas as etapas do processo de modelagem, incluindo:

* Preparação de pipelines de machine learning, com normalização, codificação e imputação;
* Treinamento simultâneo de múltiplos algoritmos com validação cruzada;
* Comparação de desempenho com base em métricas padronizadas;
* Ajuste de hiperparâmetros via métodos de otimização automática;
* Salvamento do melhor modelo com seus parâmetros, dataset de origem, métrica de referência e data de criação.

Os resultados são exibidos em tabelas interativas e gráficos visuais que facilitam a interpretação e a escolha do modelo mais adequado. A abordagem favorece a produtividade e a reprodutibilidade dos experimentos, dispensando a necessidade de codificação manual (Ali et al., 2020).

**5. Visualização de Métricas e Gráficos de Desempenho**

O módulo de visualização apresenta indicadores de performance e gráficos interativos, organizados de acordo com o tipo de tarefa:

* Classificação: matriz de confusão, curva ROC, curva de precisão-recall, acurácia, F1-score, etc.;
* Regressão: gráfico de resíduos, curva de predição, R², MAE, RMSE;
* Clusterização: score de Silhouette, distribuição de clusters, gráficos 2D/3D de agrupamentos com redução de dimensionalidade.

Esses recursos são fundamentais para avaliar a capacidade preditiva dos modelos e comunicar os resultados de forma clara, acessível e interpretável (Waskom, 2021; Géron, 2019).

**6. Predição com Novos Dados**

O sistema oferece uma interface adaptativa para predição, onde os campos de entrada são gerados dinamicamente com base nas variáveis preditoras do modelo selecionado. A lógica implementa:

* Sliders para variáveis numéricas com intervalo baseado nos valores observados;
* Seletores de categoria para atributos discretos;
* Validação de tipos e formatos durante a entrada.

Após a submissão, o modelo realiza a predição em tempo real, exibindo os resultados com interpretação contextual (ex: probabilidade por classe, valores estimados, classificação de cluster). Também é possível registrar os dados inseridos e suas respectivas previsões para análise posterior.

**7. Persistência e Gerenciamento de Metadados**

Todos os elementos do ciclo de vida do projeto — datasets, modelos, predições e configurações — são armazenados e gerenciados por meio de uma base de dados relacional estruturada com SQLAlchemy (Bayer, 2024). O sistema realiza:

* Registro automático de experimentos;
* Versionamento de modelos;
* Associação de predições a datasets e modelos específicos;
* Consulta e reuso de modelos antigos com base em data, tipo de tarefa e desempenho.

Essa abordagem segue os preceitos do MLOps e garante a governança de modelos e a auditoria das decisões baseadas em dados (Sculley et al., 2015), elementos críticos em ambientes institucionais e corporativos.

**Geração de Gráficos e Relatórios**

No contexto do sistema ML Data HUB, a geração de gráficos e relatórios configura-se como um componente central para a análise, interpretação e comunicação dos resultados produzidos em todas as fases do pipeline de ciência de dados. A visualização de dados, embasada em princípios estatísticos rigorosos e design informacional eficaz, é amplamente reconhecida como uma etapa crucial para facilitar a compreensão de padrões complexos, identificar tendências e detectar anomalias em conjuntos de dados heterogêneos e de alta dimensionalidade (Tukey, 1977; Waskom, 2021).

**Visualizações na Análise Exploratória de Dados (EDA)**

Durante a etapa inicial de Análise Exploratória de Dados (EDA), o sistema oferece um conjunto abrangente de gráficos que permitem aos usuários compreender profundamente a estrutura dos dados. São disponibilizados:

* Histogramas e gráficos de densidade, para análise da distribuição de variáveis numéricas, possibilitando a identificação de assimetrias, multimodalidades e tendências centrais;
* Gráficos de barras e de setores, para variáveis categóricas, facilitando a visualização da frequência e da proporção de categorias distintas;
* Mapas de calor para matrizes de correlação, que destacam a intensidade e direção das relações lineares entre variáveis, auxiliando na detecção de multicolinearidade e na seleção informada de preditores;
* Boxplots e violin plots, que permitem a análise visual de outliers e da dispersão dos dados;
* Gráficos de dispersão bivariados e multivariados, muitas vezes integrados a técnicas de redução de dimensionalidade (como PCA ou t-SNE) para visualização de agrupamentos naturais ou tendências não lineares.

Essas visualizações são construídas com o suporte das bibliotecas Python Matplotlib e Seaborn, conhecidas por sua robustez e flexibilidade na criação de gráficos estatísticos sofisticados (Hunter, 2007; Waskom, 2021). A seleção criteriosa do tipo de gráfico é orientada pelo tipo e distribuição dos dados, assegurando clareza e eficácia na comunicação visual (Géron, 2019).

**Relatórios de Desempenho Pós-Modelagem**

Após o processo de treinamento dos modelos de aprendizado de máquina, o sistema gera relatórios detalhados de desempenho, integrando tanto métricas quantitativas quanto gráficos explicativos, adequados para cada tarefa de modelagem:

* Para classificação, são apresentados:
  + Matriz de confusão, evidenciando verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos;
  + Curvas ROC (Receiver Operating Characteristic) e AUC (Area Under the Curve), que expressam a capacidade discriminativa do modelo em diferentes limiares;
  + Curvas de precisão-recall, particularmente úteis em cenários de classes desbalanceadas;
  + Métricas como acurácia, precisão, recall, F1-score, que permitem avaliação granular do desempenho.
* Para regressão, os relatórios incluem:
  + Gráficos de resíduos, que ajudam a identificar heterocedasticidade e viés de predição;
  + Gráficos comparativos entre valores reais e previstos, visualizando a aderência do modelo;
  + Métricas como R² (coeficiente de determinação), MAE (erro absoluto médio) e RMSE (raiz do erro quadrático médio).
* Para clusterização, são fornecidos:
  + Scores de silhueta e outros índices de avaliação da coesão e separação dos clusters;
  + Visualizações em 2D ou 3D dos agrupamentos, com técnicas de redução de dimensionalidade para facilitar a interpretação.

Esses relatórios são indispensáveis para uma avaliação crítica e fundamentada dos modelos treinados, permitindo a comparação entre algoritmos concorrentes, a identificação de potenciais limitações e a orientação para ajustes ou refinamentos futuros (Pedregosa et al., 2011; Ali et al., 2020).

**Integração Dinâmica e Comunicação Visual**

A integração desses gráficos e relatórios à interface interativa baseada em Streamlit potencializa a experiência do usuário, promovendo um ambiente de análise dinâmico, responsivo e transparente. A possibilidade de explorar visualizações em tempo real, ajustar parâmetros e comparar resultados de forma intuitiva favorece a tomada de decisões baseada em evidências.

Além disso, a apresentação visual eficaz dos resultados é um componente essencial para a comunicação dos achados científicos, facilitando a compreensão tanto por especialistas quanto por audiências menos técnicas, incluindo gestores, stakeholders e decisores (Knaflic, 2015). Essa capacidade comunicativa aumenta o impacto e a aplicabilidade prática dos projetos desenvolvidos no ML Data HUB, alinhando-se às melhores práticas do campo da ciência de dados aplicada.

**Conclusão**

O desenvolvimento do sistema ML Data HUB representa uma contribuição significativa para a democratização e a simplificação do uso de técnicas avançadas de aprendizado de máquina por meio de uma plataforma web intuitiva e integrada. Ao combinar uma interface interativa construída com Streamlit, um backend robusto fundamentado no PyCaret e uma arquitetura modular, o sistema oferece uma solução completa que abrange desde o pré-processamento e análise exploratória até a modelagem automatizada e a geração de previsões com novos dados.

Os resultados obtidos demonstram que a aplicação atende de maneira eficaz aos requisitos fundamentais da ciência de dados moderna, promovendo a reprodutibilidade dos experimentos, o versionamento dos modelos e o armazenamento estruturado de dados e metadados, aspectos essenciais para a gestão do ciclo de vida dos projetos de machine learning. A integração de relatórios detalhados e gráficos interativos fortalece a capacidade analítica e interpretativa dos usuários, ampliando o acesso ao conhecimento gerado e facilitando a tomada de decisões fundamentadas em dados.

Além disso, o uso de bibliotecas consolidadas, como PyCaret para automação da modelagem e pandas, matplotlib e seaborn para manipulação e visualização dos dados, demonstra o alinhamento do sistema às melhores práticas e tendências atuais na área, incluindo o movimento low-code/no-code que visa reduzir barreiras técnicas e acelerar o desenvolvimento de soluções inteligentes (Ali et al., 2020).

O ML Data HUB também evidencia a importância da modularidade e da adoção de boas práticas de engenharia de software no desenvolvimento de aplicações de ciência de dados, facilitando a manutenção, escalabilidade e futuras extensões do sistema. A possibilidade de incorporar novos algoritmos, metodologias de pré-processamento ou mesmo integrar outras ferramentas de análise representa um caminho promissor para o contínuo aprimoramento da plataforma.

Por fim, destaca-se que o sistema possui potencial para ser utilizado em diversos contextos aplicados, como saúde, finanças, educação e indústria, onde a análise de dados e o aprendizado de máquina vêm se mostrando indispensáveis para a inovação e otimização de processos. Futuras versões poderão contemplar melhorias na interface de usuário, suporte a dados não estruturados, explicabilidade dos modelos (por exemplo, via SHAP ou LIME) e automação ainda maior do pipeline, aproximando o sistema das demandas emergentes da ciência de dados aplicada e da inteligência artificial responsável.

Em suma, o ML Data HUB configura-se como uma ferramenta versátil e acessível, capaz de ampliar o alcance e a eficácia do aprendizado de máquina, promovendo a cultura de dados e a inteligência analítica em ambientes acadêmicos e profissionais.

### **Referências**

ALI, M. et al. PyCaret: An open source, low-code machine learning library in Python. Disponível em: [https://pycaret.org/](https://pycaret.org/). Acesso em: 10 jun. 2025.

BAYER, M. SQLAlchemy Documentation. Disponível em: [https://docs.sqlalchemy.org/](https://docs.sqlalchemy.org/). Acesso em: 14 jun. 2025.

GÉRÓN, A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 2. ed. O’Reilly Media, 2019.

HUNTER, J. D. Matplotlib: A 2D graphics environment. \*Computing in Science & Engineering\*, v. 9, n. 3, p. 90–95, 2007.

KNAFLIC, C. N. Storytelling with Data: A Data Visualization Guide for Business Professionals. Wiley, 2015.

KUHN, M.; JOHNSON, K. Feature Engineering and Selection: A Practical Approach for Predictive Models. CRC Press, 2019.

MCKINNEY, W. Data Structures for Statistical Computing in Python. In: \*Proceedings of the 9th Python in Science Conference\*, 2010. p. 51–56.

PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python. \*Journal of Machine Learning Research\*, v. 12, p. 2825–2830, 2011.

SCULLEY, D. et al. Hidden Technical Debt in Machine Learning Systems. In: \*Advances in Neural Information Processing Systems\*, v. 28, 2015.

STREAMLIT DOCUMENTATION. Disponível em: [https://docs.streamlit.io/](https://docs.streamlit.io/). Acesso em: 10 jun. 2025.

TUKEY, J. W. Exploratory Data Analysis. Addison-Wesley, 1977.

WASKOM, M. Seaborn: Statistical Data Visualization. \*Journal of Open Source Software\*, v. 6, n. 60, p. 3021, 2021.