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

Nome: Leonardo de Lima Póss RA: 22013563-2

### **Visão Geral**

Este documento apresenta o desenvolvimento de um sistema interativo voltado à simplificação do processo de criação de modelos preditivos a partir de dados tabulares. A proposta consiste em disponibilizar uma interface gráfica acessível, por meio da qual o usuário pode carregar conjuntos de dados, realizar análises exploratórias iniciais e aplicar algoritmos de modelagem estatística utilizando a biblioteca PyCaret. Essa biblioteca oferece suporte a tarefas comuns em aprendizado de máquina, como classificação, regressão e agrupamento (clustering), integrando múltiplas etapas do processo em fluxos automatizados.

A motivação central para o desenvolvimento do sistema está na tentativa de tornar mais acessível o uso de métodos computacionais aplicados à análise de dados. Embora as bibliotecas e ferramentas disponíveis atualmente permitam uma ampla gama de análises, seu uso ainda exige conhecimento prévio em linguagens de programação, especialmente Python. O sistema aqui descrito visa mitigar essa barreira técnica, oferecendo uma plataforma onde usuários podem experimentar técnicas de modelagem sem a necessidade de escrever código.

Durante o processo de desenvolvimento, buscou-se priorizar a clareza das operações realizadas e a organização do fluxo de trabalho. O sistema foi estruturado para guiar o usuário desde a etapa de ingestão dos dados até a obtenção de modelos avaliados e comparados automaticamente. Além disso, ferramentas como o YData Profiling e o D-Tale foram integradas à aplicação com o objetivo de apoiar o entendimento das características estatísticas dos dados, permitindo uma etapa exploratória rica mesmo para quem não tem familiaridade com bibliotecas como Pandas ou Seaborn.

A escolha das variáveis a serem utilizadas como preditoras e alvo é realizada de forma manual pelo usuário, por meio da interface. Essa etapa é essencial para o correto funcionamento das rotinas de treinamento automático. No caso de tarefas de agrupamento, em que não há variável alvo, o sistema realiza a modelagem diretamente a partir da matriz de dados, utilizando algoritmos como KMeans ou DBSCAN.

Outro ponto importante é o foco na transparência dos modelos gerados. O sistema apresenta, ao final do processo, informações sobre o desempenho dos modelos comparados, além de gráficos auxiliares e tabelas de métricas. Esses elementos permitem uma análise crítica sobre o comportamento dos modelos, mesmo por parte de usuários com conhecimento limitado na área.

Embora não substitua uma análise aprofundada realizada com ferramentas programáveis, o sistema pode funcionar como um recurso introdutório ao aprendizado de máquina, útil para estudantes, professores, pesquisadores e interessados em geral. Ele permite uma primeira aproximação com conceitos como treino/teste, escolha de algoritmos, overfitting, importância de variáveis, entre outros.

De modo geral, o sistema contribui para a disseminação do conhecimento em ciência de dados ao oferecer uma abordagem prática, direta e interativa para a criação de modelos preditivos. Seu desenvolvimento buscou aliar automatização a compreensibilidade, com o intuito de aproximar técnicas quantitativas de públicos com diferentes níveis de experiência.

### **Fundamentos Teóricos**

O desenvolvimento do sistema proposto está fundamentado em três pilares centrais: análise exploratória de dados automatizada (EDA), modelagem preditiva com a biblioteca PyCaret e interface interativa por meio da plataforma Streamlit. Esses pilares combinam práticas consolidadas da ciência de dados com tecnologias que favorecem a automação e a acessibilidade, formando a base teórica e prática da aplicação desenvolvida:

1. **Análise Exploratória Automatizada (EDA):**

A análise exploratória de dados (EDA – Exploratory Data Analysis) é uma das etapas mais importantes no ciclo de desenvolvimento de projetos em ciência de dados. Introduzido formalmente por John Tukey na década de 1970, o EDA tem como principal função investigar os dados de forma descritiva antes da aplicação de qualquer técnica inferencial ou preditiva (Tukey, 1977). Essa investigação inicial permite observar a estrutura dos dados, detectar padrões relevantes e identificar anomalias ou erros que possam comprometer a modelagem posterior.

Segundo Hyndman e Athanasopoulos (2018), a EDA oferece suporte valioso à compreensão do comportamento das variáveis, facilitando a construção de hipóteses e o direcionamento das etapas subsequentes do processo analítico. Entre as práticas mais comuns estão a geração de gráficos (como histogramas, boxplots e diagramas de dispersão), cálculos de medidas estatísticas (média, mediana, desvio padrão) e identificação de outliers e dados ausentes. Essas ações são fundamentais para diagnosticar características que podem influenciar o desempenho dos modelos, como variáveis desbalanceadas ou altamente correlacionadas.

Nos últimos anos, com o crescimento dos volumes de dados e da necessidade de agilizar análises, surgiram ferramentas que automatizam parte ou toda a etapa de EDA. Bibliotecas como Pandas Profiling, Sweetviz e Autoviz, por exemplo, permitem gerar relatórios completos com uma única linha de código, contendo gráficos interativos, resumos estatísticos e sugestões iniciais sobre o tratamento dos dados (Pedregosa et al., 2011; Raschka & Mirjalili, 2020). Essa automatização reduz o tempo necessário para realizar diagnósticos preliminares, especialmente em contextos com prazos curtos ou demandas recorrentes.

No sistema proposto, a análise exploratória é realizada de forma automatizada com base nesses princípios. O usuário pode carregar um conjunto de dados tabular e, imediatamente, receber uma série de visualizações e estatísticas que ajudam a interpretar as principais características do dataset. Essa abordagem é útil principalmente para usuários iniciantes ou intermediários, que desejam obter insights rápidos sem a necessidade de escrever código ou configurar bibliotecas específicas. Além disso, são destacados potenciais problemas, como colunas com baixa variabilidade, valores ausentes e discrepâncias em distribuições numéricas e categóricas.

Por fim, vale destacar que a EDA não substitui a interpretação crítica do analista, mas serve como ponto de partida para decisões mais fundamentadas. Conforme defendido por Dasu e Johnson (2003), o valor da análise exploratória reside em sua capacidade de revelar aspectos inesperados dos dados, contribuindo tanto para o ajuste de modelos quanto para o entendimento do contexto do problema analisado. Assim, ao automatizar essa etapa, o sistema desenvolvido neste trabalho busca equilibrar eficiência e rigor analítico, promovendo o uso consciente de recursos computacionais no processo de descoberta de conhecimento.

1. **Modelagem com PyCaret:**

A modelagem preditiva é uma das etapas centrais em projetos de ciência de dados, envolvendo a escolha de algoritmos, o preparo dos dados de entrada e a avaliação dos resultados gerados. Tradicionalmente, esse processo exige conhecimento técnico aprofundado em bibliotecas como Scikit-learn, XGBoost ou LightGBM, além de experiência com engenharia de atributos, validação cruzada e otimização de hiperparâmetros. Contudo, ferramentas como o PyCaret surgiram com a proposta de reduzir a complexidade operacional e acelerar a prototipação de modelos por meio de uma interface unificada e automatizada.

O PyCaret é uma biblioteca de código aberto escrita em Python que oferece suporte a tarefas de classificação, regressão, clusterização, análise de anomalias e processamento de linguagem natural. Seu objetivo principal é permitir que o usuário execute rapidamente um pipeline completo de machine learning, englobando etapas como preparação dos dados, treinamento de múltiplos algoritmos, comparação de desempenho, ajuste fino e interpretação dos resultados. Segundo Ali (2020), idealizador da biblioteca, o PyCaret visa democratizar o acesso ao aprendizado de máquina, tornando-o mais acessível para analistas, cientistas de dados iniciantes e desenvolvedores.

Um dos principais diferenciais do PyCaret é sua capacidade de encapsular operações complexas em comandos de alto nível, facilitando o uso mesmo para quem não possui domínio profundo das bibliotecas subjacentes. Em uma única função, por exemplo, o usuário pode comparar o desempenho de diversos algoritmos utilizando validação cruzada e métricas padronizadas, como acurácia, F1-score, RMSE ou R², dependendo da natureza do problema. Além disso, a biblioteca integra módulos de explicabilidade baseados em SHAP (SHapley Additive exPlanations), auxiliando na interpretação dos modelos treinados, algo cada vez mais valorizado em aplicações práticas.

A integração do PyCaret ao sistema proposto neste trabalho proporciona eficiência e robustez ao processo de modelagem automatizada, permitindo que o usuário selecione o tipo de problema (classificação, regressão ou clusterização) e obtenha rapidamente os melhores modelos disponíveis, já acompanhados por métricas e gráficos explicativos. Isso é particularmente útil em contextos de análise exploratória rápida ou validação inicial de hipóteses, quando há necessidade de avaliar múltimos modelos com baixo custo computacional e sem codificação manual extensiva.

Adicionalmente, estudos recentes têm demonstrado a relevância de bibliotecas de AutoML, como o PyCaret, no ambiente acadêmico e empresarial, sobretudo em tarefas que exigem agilidade e reprodutibilidade (He et al., 2021; Truong et al., 2019). Ao automatizar as etapas repetitivas do fluxo de trabalho, essas ferramentas liberam tempo para análises mais estratégicas, aumentando a produtividade sem comprometer a qualidade dos modelos. Dessa forma, o uso do PyCaret no sistema proposto alinha-se a uma tendência crescente na ciência de dados moderna: o equilíbrio entre automação, usabilidade e confiabilidade analítica.

1. **Interface Interativa com Upload Simples:**

A acessibilidade e a facilidade de uso são aspectos cada vez mais valorizados em sistemas voltados para análise de dados. Interfaces interativas permitem que usuários sem familiaridade com linguagens de programação possam usufruir dos benefícios da ciência de dados de forma prática e intuitiva. Nesse contexto, ferramentas como o Streamlit têm se destacado por oferecer uma abordagem direta para a criação de aplicações web baseadas em Python, com ênfase na simplicidade de desenvolvimento e na interação em tempo real com o usuário.

O Streamlit é uma biblioteca open source criada com o objetivo de transformar scripts de Python em interfaces gráficas funcionais, com o mínimo de configuração. A grande vantagem da ferramenta é permitir que cientistas de dados e analistas construam aplicações visuais sem precisar recorrer a frameworks complexos como Flask ou Dash. Em um único arquivo Python, é possível adicionar componentes como upload de arquivos, campos interativos, seletores, gráficos e até visualizações em tempo real de modelos e dados processados (TREUILLE; NIEWIADOMSKI; MALIK, 2019). Essa abordagem facilita a prototipação rápida de sistemas voltados para exploração e modelagem de dados.

No sistema desenvolvido neste trabalho, o Streamlit foi utilizado para criar uma interface limpa e funcional, na qual o usuário pode realizar o upload de arquivos de dados tabulares (como arquivos .csv) e iniciar o processo completo de análise exploratória e modelagem com poucos cliques. A interface organiza as seções em abas ou etapas sequenciais, orientando o usuário desde o envio dos dados até a interpretação dos resultados. Isso não apenas reduz a curva de aprendizado, como também evita erros comuns relacionados à manipulação manual de código ou dependências externas.

A literatura atual reforça a importância de interfaces simples e interativas na disseminação de ferramentas de ciência de dados. De acordo com Kluyver et al. (2016), o uso de notebooks interativos e interfaces gráficas facilita o aprendizado, promove reprodutibilidade e amplia o acesso a técnicas computacionais. O Streamlit segue essa mesma filosofia, possibilitando que análises sofisticadas sejam executadas e compreendidas por um público mais amplo. Dessa forma, sua utilização no presente trabalho colabora para tornar a análise preditiva mais acessível, especialmente em ambientes educacionais e de pesquisa aplicada.

### **Funcionalidades**

O sistema proposto foi desenvolvido com o objetivo de simplificar e democratizar o processo de análise e modelagem de dados tabulares, por meio de uma interface interativa e intuitiva, voltada para usuários de diversos níveis de experiência, incluindo aqueles sem conhecimentos avançados em programação ou ciência de dados. A plataforma integra um conjunto robusto de funcionalidades que englobam desde a ingestão automatizada de dados até a geração de modelos preditivos e análise textual, promovendo um fluxo contínuo e coerente para exploração, transformação e interpretação dos dados.

Dentre as funcionalidades centrais, destaca-se a capacidade de ingestão de dados a partir de links externos, contemplando suporte direto aos principais repositórios públicos de dados, como o Kaggle e o HuggingFace. O sistema é capaz de identificar automaticamente a origem do dataset a partir da URL fornecida, acionando os respectivos módulos de download e descompactação. Para o Kaggle, é realizada a autenticação via API, seguida do download e extração dos arquivos no formato original, com tratamento para múltiplos formatos comuns, como CSV, JSON, XLSX e ZIP. Já para a Hugging Face, o sistema utiliza a biblioteca oficial datasets para carregar os dados diretamente, convertendo-os para formatos manipuláveis, como DataFrames pandas, e armazenando-os localmente. Durante esse processo, mensagens informativas e alertas são exibidos em tempo real na interface, permitindo ao usuário acompanhar o progresso e identificar eventuais falhas ou restrições, como problemas de autenticação, limitação de quota ou formatos incompatíveis.

Após a ingestão dos dados, a plataforma disponibiliza ferramentas avançadas para análise exploratória, fundamentais para compreender a estrutura, qualidade e padrões presentes no conjunto de dados. São oferecidas duas abordagens complementares: o D-Tale, que cria uma interface web interativa para navegação e manipulação dinâmica das tabelas, com filtros customizáveis, ordenação, visualização de tipos e contagem de valores nulos; e o YDataProfiling (anteriormente conhecido como pandas-profiling), que gera relatórios estatísticos detalhados contendo descrições univariadas, distribuições, matrizes de correlação (Pearson, Spearman, etc.), detecção de outliers, além de alertas automáticos para dados ausentes, valores constantes, dados duplicados ou possíveis problemas de qualidade. Essa combinação permite uma análise inicial robusta e multifacetada, auxiliando o usuário a tomar decisões fundamentadas para a etapa seguinte.

Na sequência, o sistema conduz o usuário pelo processo de configuração da tarefa de modelagem preditiva. A interface permite a seleção explícita das variáveis independentes e da variável dependente (target) em tarefas supervisionadas, contemplando problemas de classificação e regressão. Alternativamente, o usuário pode optar por uma tarefa de clusterização, para análise não supervisionada, em que o target não é requerido. A plataforma valida essas escolhas para garantir coerência e integridade dos dados, prevenindo erros comuns, como a seleção de colunas incompatíveis ou ausência de variáveis-alvo. Esse fluxo guiado reduz a complexidade para usuários menos experientes, ao mesmo tempo que mantém flexibilidade para usuários avançados ajustarem suas análises.

A etapa de modelagem utiliza a biblioteca PyCaret, que automatiza diversos processos fundamentais do ciclo de vida de machine learning. Internamente, a plataforma executa rotinas de pré-processamento de dados, incluindo tratamento de valores ausentes, codificação de variáveis categóricas, normalização e seleção automática de recursos, por meio da função setup(). Posteriormente, o sistema aplica a função compare\_models(), que realiza uma busca comparativa entre múltiplos algoritmos padrão da área, tais como Random Forest, XGBoost, LightGBM para tarefas supervisionadas e KMeans, DBSCAN, HDBSCAN para clusterização. Opcionalmente, o usuário pode ativar processos de tuning automático, que ajustam hiperparâmetros para otimizar a performance dos modelos, além de visualizar gráficos e métricas relevantes, como acurácia, AUC, RMSE e silhueta , gerados pela função plot\_model(). Os resultados são apresentados de forma clara e visualmente acessível, facilitando a compreensão do desempenho e comportamento dos modelos, mesmo para quem não possui expertise em machine learning.

Adicionalmente, a plataforma incorpora módulos de análise textual, focados em análise de sentimentos e classificação de textos. O usuário pode realizar análises básicas, como contagem de palavras, frequência e visualização de n-gramas, ou utilizar modelos pré-treinados para classificação de sentimentos e categorias textuais. Essa funcionalidade é particularmente útil em cenários que envolvem dados qualitativos, como avaliações, comentários e feedbacks. As análises textuais podem ser integradas ao pipeline geral, enriquecendo as possibilidades de investigação e extração de insights.

Para garantir a persistência e rastreabilidade das análises, o sistema permite o salvamento manual dos datasets e modelos treinados em banco de dados local, facilitando a reutilização e auditoria dos processos. Essa característica também contribui para a organização dos projetos e facilita a colaboração em ambientes acadêmicos e corporativos.

Em suma, a plataforma configura-se como uma solução completa e integrada para a análise exploratória, modelagem preditiva automatizada e análise textual, voltada para o ambiente acadêmico e de prototipação rápida. A simplicidade da interface combinada com a robustez das ferramentas subjacentes permite que usuários de diferentes perfis acessem recursos avançados de ciência de dados sem barreiras técnicas, promovendo uma cultura de decisão orientada a dados com maior agilidade e confiança.

### **Roadmap de Desenvolvimento**

|  |  |  |
| --- | --- | --- |
| **Etapa** | **Funcionalidade** | **Justificativa** |
| Versão 1 | Integração com a interface  Streamlit | Facilitar a visualização do procedimento. |
| Versão 2 | Inclusão da funcionalidade que possibilita a escolha de datasets | Aumento da versatilidade do projeto |
| Versão 3 | Inclusão de funcionalidades EDA com visualização integrado no Streamlit | Aumentar a interpretabilidade dos dados em um ambiente único |
| Versão 4 | Inclusão da opção de baixar datasets, via link da plataforma kaggle | Facilitar a integração com novos dados |
| Versão 5 | Inclusão da opção de baixar datasets, via link da plataforma Hugging Face | Aumentar ainda mais a possibilidade de variação dos dados |

### **Conclusão**

O sistema proposto configura-se como uma ferramenta interativa voltada à análise, modelagem e interpretação de dados tabulares, com foco na integração de múltiplas bibliotecas e práticas comuns em ciência de dados dentro de um único ambiente gráfico. Sua concepção buscou unir acessibilidade e robustez, de modo a atender tanto usuários iniciantes quanto aqueles com experiência intermediária ou avançada na área. A interface desenvolvida facilita o fluxo completo de trabalho, desde a ingestão de dados até a geração de modelos preditivos, com a vantagem de dispensar a escrita de código, tornando-se adequada para contextos educacionais, exploratórios ou de prototipagem.

Um dos principais diferenciais do sistema está na forma como organiza e automatiza etapas fundamentais do pipeline de ciência de dados. A preparação dos dados, incluindo tratamento de valores ausentes, detecção de outliers e codificação de variáveis, é realizada automaticamente por meio da biblioteca PyCaret, reduzindo o risco de inconsistências causadas por configurações manuais mal definidas. Em seguida, o sistema conduz a seleção e comparação entre diferentes algoritmos, utilizando métricas padronizadas para apresentar ao usuário os modelos com melhor desempenho. Essa abordagem não apenas reduz a carga técnica, como também facilita a compreensão dos trade-offs entre os modelos avaliados.

Além do aprendizado supervisionado, o sistema suporta tarefas de aprendizado não supervisionado, como agrupamento de instâncias com base em similaridade. Nesses casos, o sistema recorre a algoritmos como KMeans, DBSCAN e HDBSCAN, permitindo ao usuário explorar estruturas subjacentes nos dados sem a necessidade de definir um alvo. Tal funcionalidade é especialmente útil em análises exploratórias de dados desconhecidos ou na busca por padrões em domínios como marketing, ciências sociais e estudos comportamentais.

Outro aspecto relevante do sistema é seu compromisso com a explicabilidade e a transparência dos modelos gerados. Inspirado nos princípios descritos por Doshi-Velez e Kim (2017), o sistema disponibiliza visualizações como gráficos de importância de variáveis, curvas de aprendizado e matrizes de confusão, além de métricas padronizadas (ex.: AUC, R², RMSE, Silhouette Score) que ajudam na interpretação dos resultados. Tais recursos são fundamentais para garantir que o processo de modelagem possa ser auditado e compreendido, o que é particularmente importante em contextos que exigem rastreabilidade e prestação de contas, como pesquisas científicas e estudos de impacto.

A plataforma também promove o acesso facilitado a bases de dados públicas. Através da integração com repositórios como Kaggle e Hugging Face, o usuário pode importar datasets diretamente a partir de URLs, eliminando a necessidade de downloads manuais ou scripts auxiliares. O sistema reconhece o domínio de origem e executa os comandos necessários para autenticação, extração e armazenamento dos arquivos em diretórios internos previamente configurados. Essa funcionalidade tem como objetivo acelerar o processo de experimentação, contribuindo para uma experiência mais fluida na análise inicial dos dados.

No que tange à análise exploratória, o sistema oferece duas ferramentas complementares: o YData Profiling, voltado para a geração de relatórios estatísticos automatizados, e o D-Tale, que provê uma interface dinâmica para exploração manual dos dados. O YData Profiling realiza tarefas como análise univariada, avaliação de correlações, geração de histogramas e identificação de possíveis inconsistências, enquanto o D-Tale permite inspeções detalhadas por meio de filtros, buscas e visualizações interativas. Juntas, essas abordagens contribuem para a formação de um panorama sólido sobre a natureza e estrutura do conjunto de dados antes do início da modelagem.

A seleção de variáveis preditoras e da variável alvo (quando aplicável) é conduzida de maneira interativa pelo usuário, garantindo flexibilidade na definição do escopo da modelagem. Em tarefas de classificação e regressão, essa etapa é essencial para garantir que a biblioteca PyCaret possa configurar corretamente os parâmetros de entrada, aplicar transformações e validar os modelos por meio de partições de treino e teste. No caso de clusterização, a modelagem é realizada diretamente sobre o conjunto de atributos, permitindo ao sistema identificar agrupamentos latentes sem supervisão.

A arquitetura do sistema também contempla a persistência de modelos treinados, utilizando uma abordagem baseada em banco de dados SQLite para o armazenamento dos artefatos gerados. Através de serialização em formato binário (pickle), os modelos podem ser salvos, recuperados e aplicados posteriormente em novos dados. Essa funcionalidade é particularmente útil em contextos que envolvem comparação entre experimentos, continuidade de projetos ou aplicação prática dos modelos em sistemas externos.

Por fim, o sistema ainda oferece um módulo complementar voltado para análise de sentimentos e classificação textual, demonstrando sua flexibilidade em lidar com diferentes tipos de dados. Embora não seja o foco central da proposta, esse módulo estende as possibilidades de aplicação da ferramenta para contextos envolvendo linguagem natural, como análises de avaliações de usuários, comentários ou posts em redes sociais.

Em conjunto, todos esses elementos constituem um ambiente unificado e coerente para análise e modelagem de dados, promovendo a compreensão dos processos envolvidos em machine learning e democratizando o acesso a técnicas quantitativas que, tradicionalmente, requerem habilidades mais avançadas de programação. Ao integrar funcionalidades úteis com uma interface intuitiva, o sistema representa uma contribuição relevante tanto para atividades didáticas quanto para aplicações exploratórias e desenvolvimento de provas de conceito em ciência de dados.

### **Referências**

**ALI, Moez.** PyCaret: an open-source, low-code machine learning library in Python. 2020. Disponível em: https://pycaret.org/. Acesso em: 16 jun. 2025.

**BISHOP, Christopher M.** *Pattern recognition and machine learning*. New York: Springer, 2006.

**DASU, Tamraparni; JOHNSON, Theodore.** *Exploratory data mining and data cleaning*. Hoboken: John Wiley & Sons, 2003.

**DOSHI-VELEZ, Finale; KIM, Been.** Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

**FEURER, Matthias et al.** AutoML: methods, systems, challenges. In: *Automated Machine Learning*. Springer, Cham, 2020. p. 1–54.

**HE, Xin et al.** AutoML: a survey of the state-of-the-art. *Knowledge-Based Systems*, v. 212, 2021. Disponível em: https://doi.org/10.1016/j.knosys.2020.106622. Acesso em: 16 jun. 2025.

**Hyndman, Rob J.; Athanasopoulos, George.** *Forecasting: principles and practice*. 2. ed. Melbourne: OTexts, 2018. Disponível em: https://otexts.com/fpp2/. Acesso em: 16 jun. 2025.

**KLUYVER, Thomas et al.** Jupyter Notebooks – a publishing format for reproducible computational workflows. In: LOIZIDES, Fernando; SCHMIDT, Birgit (ed.). *Positioning and power in academic publishing: players, agents and agendas*. Amsterdam: IOS Press, 2016. p. 87–90. Disponível em: https://ebooks.iospress.nl/volumearticle/42900. Acesso em: 16 jun. 2025.

**KLUYVER, Thomas et al.** Jupyter Notebooks as a publishing format for reproducible workflows. *2019*.

**NORMAN, Donald A.** *The design of everyday things*. New York: Basic Books, 2020.

**PEDREGOSA, Fabian et al.** Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Disponível em: http://jmlr.org/papers/v12/pedregosa11a.html. Acesso em: 16 jun. 2025.

**RASCHKA, Sebastian; MIRJALILI, Vahid.** *Python machine learning*. 3. ed. Birmingham: Packt Publishing, 2020.

**RICHARDSON, Leonard; RUBY, Sam.** *RESTful Web APIs*. 2. ed. Beijing: O’Reilly Media, 2017.

**TREUILLE, Adrien; NIEWIADOMSKI, Thiago; MALIK, Amanda.** Streamlit: the fastest way to build and share data apps. Streamlit Inc., 2019. Disponível em: https://streamlit.io/. Acesso em: 16 jun. 2025.

**TRUONG, Andrew D. et al.** Towards automated machine learning: evaluation and comparison of AutoML approaches and tools. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019. p. 269–278.

**TUKEY, John W.** *Exploratory data analysis*. Reading: Addison-Wesley, 1977.

**ZHAO, Wei et al.** Streamlining ML deployment with rapid prototyping. *2019*.