



**Projeto – Mini-aula em Jupyter Notebook sobre Técnicas de Aprendizado de Máquina**

**Entrega: 14/12/2025 23h59**

**Objetivo:**

Criar uma mini-aula em formato de notebook Jupyter abordando uma técnica de Aprendizado de Máquina, incluindo explicação teórica, implementação prática em Python e análise crítica dos resultados. O objetivo é avaliar a capacidade dos alunos de estudar um método novo, aplicá-lo em dados reais ou simulados e comunicá-lo de forma clara e estruturada.

**Instruções:**

- O trabalho poderá ser realizado em duplas;
- A nota deste trabalho corresponde a 100% da avaliação da Unidade III;
- Cada aluno/dupla deverá escolher uma técnica de Aprendizado de Máquina, listada neste documento, e preparar uma mini-aula sobre a mesma utilizando como recurso principal um notebook Jupyter;
  - O notebook Jupyter deverá ser desenvolvido em Python, utilizando, preferencialmente, as bibliotecas: `pandas`, `numpy`, `matplotlib`, `seaborn` e `scikit-learn`;
- A submissão do trabalho deverá ocorrer via SIGAA até a data indicada acima.
  - Deverão ser submetidos os arquivos que permitam a execução do projeto (arquivo `.ipynb` do notebook), bem como eventuais arquivos adicionais, necessários para reprodutibilidade (datasets pequenos, scripts de apoio, instruções de download se o dataset for externo);
- Após a submissão, o trabalho deverá ser apresentado presencialmente, utilizando o próprio notebook como material de apoio, em horário a ser agendado entre os dias 15/12/2025 e 16/12/2025 (conforme agenda a ser disponibilizada pelo professor, em local a definir);
- No dia da apresentação, é responsabilidade de cada aluno/dupla garantir:
  - Ambiente configurado (bibliotecas instaladas, *dataset* disponível, *kernel* funcionando);
  - Notebook executável, sem erros;
  - Organização do tempo de apresentação (aprox. 10 a 15 minutos).

## Descrição geral do projeto:

Pretende-se que cada aluno/dupla escolha uma técnica de Aprendizado de Máquina e prepare uma mini-aula em notebook Jupyter.

Essa mini-aula deverá combinar:

- Uma introdução intuitiva à técnica;
- Um resumo dos fundamentos teóricos essenciais;
- Uma aplicação prática utilizando um conjunto de dados real;
- Avaliação do desempenho do modelo e comparação com algum método já visto na disciplina (opcional);
- Uma discussão sobre usos, limitações e boas práticas relacionadas à técnica estudada.

Ao final, o notebook servirá tanto como relatório técnico quanto como roteiro de aula, sendo apresentado oralmente ao professor.

## Forma de avaliação:

- Cada aluno/dupla deverá apresentar e explicar a técnica escolhida, o conjunto de dados utilizado, as etapas do experimento e os resultados obtidos;
- O trabalho deverá estar completo e funcional no momento da apresentação, evitando-se a necessidade de instalações, configurações ou correções em tempo real;
- Serão considerados os quesitos listados abaixo, incluindo clareza da mini-aula, correção técnica, qualidade dos experimentos e da apresentação oral.

## Quesitos a serem avaliados:

- Clareza e organização da mini-aula no notebook Jupyter;
- Correção técnica da explicação teórica e da implementação do modelo;
- Qualidade dos experimentos, incluindo:
  - Escolha do *dataset*;
  - Estratégia de treino e teste (ou validação cruzada);
  - Avaliação com métricas coerentes com o tipo de problema;
- Análises e interpretações dos resultados, indo além de apenas relatar valores numéricos;
- Qualidade dos gráficos e visualizações (legibilidade, títulos, rótulos, legendas);
- Qualidade da apresentação oral, incluindo tempo, divisão de fala (no caso de dupla), clareza na explicação e domínio do conteúdo;
- Originalidade e aprofundamento, seja na escolha do *dataset*, nos experimentos realizados ou na forma de apresentar a técnica (ex: variação de hiperparâmetros, análises adicionais, pequenos experimentos extras).
  - Trabalhos que apresentarem elementos adicionais — como análises extras de hiperparâmetros, comparações mais ricas com modelos de referência, visualizações bem elaboradas ou pequenos protótipos (por exemplo, em Streamlit) — poderão receber pontuação diferenciada no quesito “Originalidade e aprofundamento”.

## Tarefas:

### Tarefa 1. Escolha e validação da técnica

A escolha da técnica deve ser feita a partir da lista abaixo. Cada aluno/dupla deverá escolher uma das opções.

- Modelagem preditiva (supervisionado):

- a) k-Nearest Neighbors (k-NN) – classificação e regressão

Técnica baseada na ideia de que exemplos parecidos tendem a ter respostas parecidas: para prever a saída de um novo ponto, o modelo olha para seus k vizinhos mais próximos no conjunto de treino. No projeto, deve ser apresentada a intuição geral do método e, em seguida, ilustrado com um exemplo de classificação e um exemplo de regressão.

- b) Naive Bayes

Classificação probabilística baseada no teorema de Bayes e na suposição de independência (ingênua) entre as features.

- c) Perceptron (classificador linear simples)

Modelo de classificação linear histórico, adequado para discutir separabilidade linear, fronteiras de decisão e limitações.

- d) Support Vector Machines (SVM) com SVC – classificação

Foco em problemas de classificação, apresentando a ideia de margens máximas, fronteiras de decisão e uso de kernels.

- e) Support Vector Machines (SVM) com SVR – regressão

Foco em regressão, explicando o conceito de margem de tolerância ( $\epsilon$ ), erro tolerado e como o modelo busca ajustar uma função que se mantenha dentro de um “tubo” ao redor dos dados.

- f) Multi-Layer Perceptron (MLP) – classificação

Rede neural rasa aplicada a problemas de classificação, utilizando a implementação MLPClassifier (scikit-learn).

- g) Multi-Layer Perceptron (MLP) – regressão

Rede neural rasa aplicada a problemas de regressão, utilizando a implementação MLPRegressor (scikit-learn).

- h) Gradient Boosting – classificação

Método de ensemble baseado em boosting de árvores de decisão, em que modelos fracos são treinados sequencialmente para melhorar o desempenho em problemas de classificação.

- i) Gradient Boosting – regressão

Mesmo princípio de boosting, porém aplicado a problemas de regressão, combinando várias árvores de decisão para aproximar uma função contínua.

- Modelagem descritiva (não supervisionado):

- j) DBSCAN

- Método de agrupamento baseado em densidade, capaz de identificar clusters de formatos arbitrários e detectar pontos de ruído/outliers.

- k) Gaussian Mixture Models (GMM)

- Técnica de agrupamento baseada em mistura de distribuições gaussianas, permitindo interpretar clusters de forma probabilística.

- l) K-Medoids

- Variante do K-Means que utiliza medoides (pontos reais do conjunto de dados) em vez de médias, sendo mais robusta a outliers (requer biblioteca scikit-learn-extra).

- m) Kernel PCA

- Extensão não linear do PCA tradicional, utilizando funções de kernel para projetar os dados em um espaço de maior dimensão e, em seguida, realizar a decomposição em componentes principais. É adequada para dados com estruturas não lineares.

- n) t-SNE

- Técnica de redução de dimensionalidade não linear com foco em visualização em 2D/3D, muito usada para explorar dados de alta dimensão, preservando a estrutura local dos vizinhos.

## Tarefa 2. Seleção e análise inicial do conjunto de dados

Escolher um *dataset* apropriado para demonstrar o funcionamento da técnica, preferencialmente de fontes públicas (repositórios conhecidos).

Descrever brevemente o conjunto de dados: número de instâncias, variáveis principais, variável-alvo (quando houver) e tipo de problema (classificação, regressão etc.).

Realizar uma análise exploratória inicial, incluindo:

- Visualização de amostras do *dataset* (`head()`);
- Estatísticas descritivas;
- Gráficos, conectando com os conteúdos de análise estatística e visualização vistos na disciplina (medidas descritivas, histogramas, *scatterplots*, etc.).

## Tarefa 3. Elaboração do notebook Jupyter (estrutura mínima da mini-aula)

Organizar o notebook de forma didática, contemplando, no mínimo, as seguintes seções:

- A. Identificação

- Título da técnica e da mini-aula;
  - Nome(s) do(s) aluno(s), matrícula(s);
  - Disciplina, curso e período.
- B. Introdução e motivação
- Explicar, em linguagem simples, que tipo de problema a técnica resolve e em que contextos é utilizada.
- C. Fundamentos teóricos básicos
- Apresentar a ideia principal da técnica (intuição);
  - Incluir expressões matemáticas comentadas, relacionando com conceitos já vistos (distância, regressão, hiperplano, etc.).
- D. Descrição do *dataset*
- Explicar as variáveis (*features*) e o alvo;
  - Comentar possíveis desafios dos dados (desbalanceamento, ruído, etc.).
- E. Preparação dos dados
- Separação treino/teste ou uso de validação cruzada;
  - Normalização/padronização quando necessário;
  - Tratamento de valores ausentes, se existirem.
- F. Implementação do modelo
- Criação do modelo com a biblioteca escolhida (scikit-learn);
  - Explicação dos principais hiperparâmetros utilizados;
  - Treinamento e geração de previsões.
- G. Avaliação do modelo
- Cálculo de métricas adequadas ao tipo de problema (acurácia, precisão, recall, F1, MAE, MSE,  $R^2$ , etc.);
  - Visualização de resultados (matriz de confusão, curvas, gráficos de clusters, etc.).
- H. Discussão e limitações
- Comentar em que situações a técnica tende a funcionar bem ou mal;
  - Discutir a sensibilidade a hiperparâmetros e ao pré-processamento.
- I. Conclusões
- Resumir o que foi aprendido sobre a técnica, os principais resultados obtidos e quando recomendariam seu uso.
- J. Referências
- Listar fontes utilizadas (documentações, artigos, livros, tutoriais confiáveis).

#### **Tarefa 4. Implementação, experimentos e análise de resultados**

Executar o notebook garantindo que todas as células rodem sem erros, de cima a baixo, em um ambiente padrão.

Realizar experimentos suficientes para ilustrar bem a técnica, como:

- Variação de hiperparâmetros principais;
- Comparação de desempenho em diferentes configurações;
- Testes com subconjuntos de dados, quando pertinente.

Interpretar os resultados obtidos, conectando as métricas e os gráficos com o comportamento esperado da técnica e com os conceitos vistos em aula (*overfitting*, *underfitting*, impacto da normalização, etc.).

#### **Tarefa 5. Preparação da apresentação (defesa oral)**

Organizar um roteiro de apresentação com duração aproximada de 10 a 15 minutos, utilizando o notebook como material principal.

Estruturar a fala abordando:

- Motivação e escolha da técnica;
- Visão geral teórica (intuição);
- Descrição do *dataset*;
- Demonstração resumida das principais etapas do notebook (pré-processamento, treinamento, avaliação);
- Principais resultados e conclusões.

No caso de trabalho em dupla, dividir a apresentação de forma equilibrada, garantindo que ambos os integrantes apresentem partes relevantes do trabalho.

Preparar-se para responder perguntas sobre:

- Funcionamento da técnica e aplicações/ usos;
- Justificativa das escolhas de modelo, hiperparâmetros e métricas;
- Interpretação dos gráficos e resultados numéricos.

### Lista de sorteio dos temas:

Cada aluno recebeu um tema sorteado aleatoriamente, conforme lista abaixo. O sorteio foi feito de forma a distribuir os temas de maneira equilibrada entre a turma, permitindo repetições (ou seja, mais de uma dupla poderá trabalhar o mesmo assunto).

Em seguida, vocês deverão se organizar em duplas, por livre escolha. Quando a dupla estiver formada, ela passará a ter dois temas disponíveis e a dupla deverá decidir, em comum acordo, qual desses temas será utilizado para preparar a aula.

É importante destacar que a dupla só poderá escolher um tema que tenha sido sorteado para pelo menos um de seus membros. Não será permitido trocar temas com outras duplas, nem escolher temas que não estejam entre os dois originalmente sorteados para os integrantes.

**Após a definição da dupla e do tema escolhido, deverão enviar os nomes dos integrantes e, também, o tema para o professor.**

Discente	Tema sorteado
ALAN CESAR REBOUCAS DE ARAUJO CARVALHO	h) Gradient Boosting – classificação
ALLYSON MATHEUS GUEDES DE OLIVEIRA	f) Multi-Layer Perceptron (MLP) – classificação
AMADEU CLEMENTINO ARAÚJO NETO	a) k-Nearest Neighbors (k-NN) – classificação e regressão
ANDRE LUIZ LIMA SOUZA	e) Support Vector Machines (SVM) com SVR – regressão
ARTHUR PHELIPE DE OLIVEIRA QUEIROZ	m) Kernel PCA
CARLOS SILVANO DE OLIVEIRA JUNIOR	h) Gradient Boosting – classificação
DANIEL BRUNO TRINDADE DA SILVA	k) Gaussian Mixture Models (GMM)
DIEGO MATHEWS GOMES DA SILVA	i) Gradient Boosting – regressão
EDIVELTON RAFAETT SILVA DE ARAÚJO	k) Gaussian Mixture Models (GMM)
EDUARDO MURILO PINTO TABORDA	n) t-SNE
ERICK HENRIQUE DA SILVA PAZ	d) Support Vector Machines (SVM) com SVC – classificação
ERIVELTO FLORENCIO DE LIMA	c) Perceptron (classificador linear simples)
FELIPE GABRIEL BEZERRA DA SILVA	f) Multi-Layer Perceptron (MLP) – classificação
FELIX LUIZ GARCAO FILHO	a) k-Nearest Neighbors (k-NN) – classificação e regressão
GABRIEL CRUZ DE LAVOR	i) Gradient Boosting – regressão
GUSTAVO PEREIRA DE CARVALHO	a) k-Nearest Neighbors (k-NN) – classificação e regressão
HÉLIO ARRUDA CÂMARA NETO	c) Perceptron (classificador linear simples)
HENRIQUE EDUARDO COSTA DA SILVA	j) DBSCAN
HUGO HENRIQUE DE VASCONCELOS FIGUEIREDO	f) Multi-Layer Perceptron (MLP) – classificação
IAGO RAFAEL RAMOS CANTUARIA MOREIRA	k) Gaussian Mixture Models (GMM)
IGOR CIRNE BORGES DE OLIVEIRA	g) Multi-Layer Perceptron (MLP) – regressão
ISRAEL SOARES DE CASTRO FILHO	n) t-SNE
JOÃO VICTOR DO NASCIMENTO RIBEIRO DE ANDRADE	b) Naive Bayes
JULIANE DA SILVA SANTOS	l) K-Medoids
LUCAS AUGUSTO SPINOLA PINTO	e) Support Vector Machines (SVM) com SVR – regressão
LUCAS MARQUES DOS SANTOS	j) DBSCAN

LUIZ EDUARDO FERNANDES LOBATO	g) Multi-Layer Perceptron (MLP) – regressão
MARIA EDUARDA SOUSA DO NASCIMENTO	m) Kernel PCA
MARIA HELENA FERNANDES LEOCÁDIO	e) Support Vector Machines (SVM) com SVR – regressão
MATEUS ANSELMO DO NASCIMENTO	j) DBSCAN
MATHEUS BEZERRA DANTAS SARAIVA	h) Gradient Boosting – classificação
MATHEUS IGOR PINHEIRO FERNANDES	i) Gradient Boosting – regressão
MATHEUS SILVA MENDES	m) Kernel PCA
NORMANDO DE ALMEIDA RODRIGUES	g) Multi-Layer Perceptron (MLP) – regressão
PEDRO HENRIQUE DA SILVA PAIXAO	d) Support Vector Machines (SVM) com SVC – classificação
PEDRO HENRIQUE DA SILVA SANTOS	n) t-SNE
RENAN DE AQUINO PEREIRA	c) Perceptron (classificador linear simples)
THAÍS KAROLAYNE MILITÃO DE LIMA	h) Gradient Boosting – classificação
VICTOR MAXIMO COSTA LIMA DE MOURA	b) Naive Bayes
VINICIUS ALEXANDRINO DE SOUZA	d) Support Vector Machines (SVM) com SVC – classificação
VIVIANE STEPHANE PINHEIRO NOVO	l) K-Medoids
WILLIANE FERREIRA CARDOSO	e) Support Vector Machines (SVM) com SVR – regressão