Universidade Federal de Pernambuco
Inteligência Artificial

Matheus Miranda Cabral de Menezes

PIPELINE COMPLETO DE PRÉ-PROCESSAMENTO

Sexta Atividade de Ciência de Dados (CIN0208)
Profa. Juscimara Gomes

Dataset Escolhido: Hepatitis
Link do Notebook Jupyter Python

Recife - PE
2025

1. INTRODUÇÃO

Objetivo da Atividade: O objetivo deste trabalho é construir, executar e avaliar um pipeline completo de pré-processamento de dados, analisando o impacto da combinação de diferentes técnicas (Imputação, Scaling, Balanceamento e Seleção de Features) no desempenho do algoritmo K-Nearest Neighbors (KNN).

Características do Dataset: O dataset utilizado foi o "Hepatitis", obtido da plataforma OpenML. Este conjunto de dados contém informações médicas de pacientes com hepatite, com o objetivo de prever se o paciente viverá ou morrerá.

- O dataset possui 155 amostras e 19 features, cumprindo o requisito de no mínimo 10 atributos.

- Dados Faltantes: Uma característica crítica deste dataset é a presença significativa de valores ausentes em diversas colunas, o que o torna ideal para testar estratégias de imputação.

- Target: A variável-alvo é a classe Class (LIVE/DIE), caracterizando um problema de classificação binária.

Protocolo Experimental: Foram testadas todas as combinações possíveis entre as técnicas listadas, totalizando 16 experimentos. A validação foi feita através de Stratified K-Fold Cross-Validation com k=5.

2. METODOLOGIA E IMPLEMENTAÇÃO

2.1 Pipeline e Ferramentas

O processo experimental foi conduzido usando Python e as bibliotecas scikit-learn e imbalanced-learn. O uso da biblioteca imbalanced-learn foi fundamental para permitir a inclusão de etapas de resampling dentro do pipeline, garantindo que o balanceamento ocorresse apenas nos folds de treino, evitando vazamento de dados.

Preparação dos Dados:

➢ O dataset foi carregado e os valores "?" foram tratados como np.nan.

Definição das Etapas do Pipeline:

1. Imputação (Preenchimento de Nulos):
    - Mediana: SimpleImputer(strategy = 'median').
    - KNNImputer: Preenchimento baseado na vizinhança (k = 5).

2. Scaling (Normalização):
    - StandardScaler: Padronização baseada em média e desvio padrão.
    - MinMaxScaler: Normalização para o intervalo [0, 1].

3. Balanceamento:
    - Nenhum: Baseline.
    - RandomOverSampler: Duplicando aleatoriamente exemplos da classe minoritária.

4. Seleção de Atributos:
    - Nenhuma: Mantém todas as 19 features.
    - SelectKBest: Seleciona as 10 melhores features baseadas no teste ANOVA (f_classif).

Avaliação:

- Para cada um dos 16 pipelines, foi calculado o **F1-Score** médio dos 5 folds e o desvio padrão. O F1-Score Weighted foi escolhido por ser uma métrica robusta para lidar com o desequilíbrio de classes inerente ao problema médico.

3. RESULTADOS

3.1 Tabela Consolidada

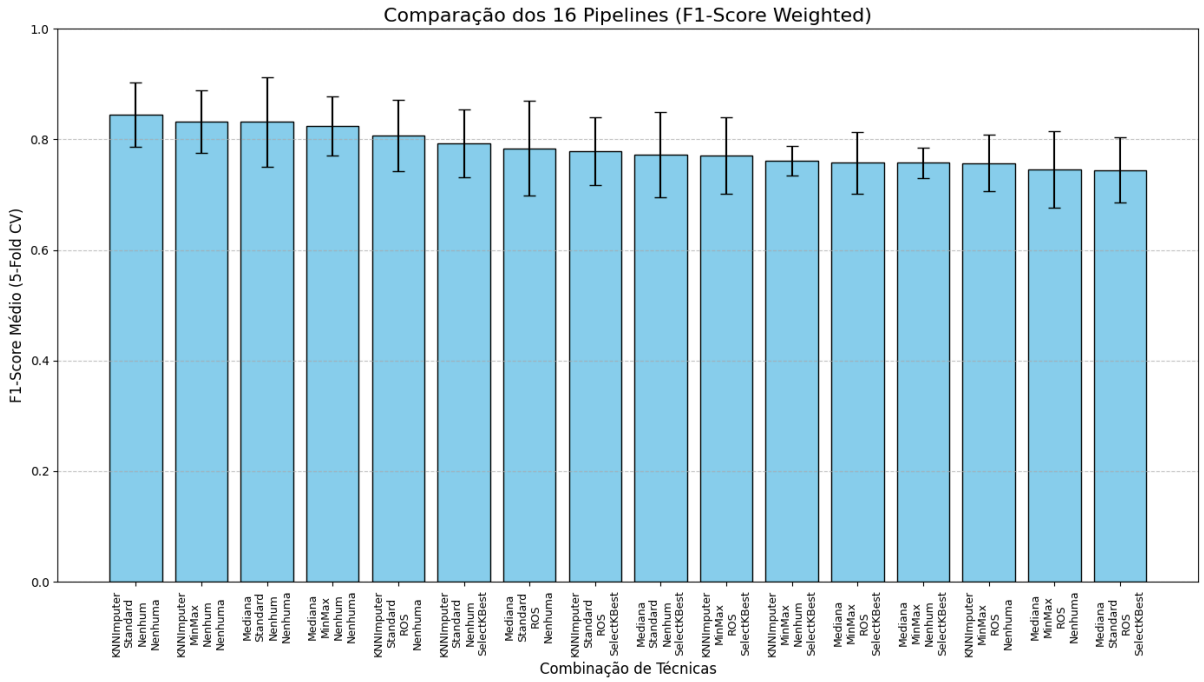
Abaixo estão os resultados consolidados dos 16 experimentos, ordenados do melhor para o pior desempenho médio:

Tabela Consolidada dos Resultados

ID	Imputação	Scaling	Balanceamento	Seleção	F1-Score Médio	Desvio Padrão
9	KNNImputer	Standard	Nenhum	Nenhuma	0.8448	0.0578
13	KNNImputer	MinMax	Nenhum	Nenhuma	0.8321	0.0573
1	Mediana	Standard	Nenhum	Nenhuma	0.8317	0.0809
5	Mediana	MinMax	Nenhum	Nenhuma	0.8245	0.0541
11	KNNImputer	Standard	ROS	Nenhuma	0.8070	0.0651
10	KNNImputer	Standard	Nenhum	SelectKBest	0.7932	0.0611
3	Mediana	Standard	ROS	Nenhuma	0.7839	0.0856
12	KNNImputer	Standard	ROS	SelectKBest	0.7785	0.0618
2	Mediana	Standard	Nenhum	SelectKBest	0.7725	0.0770
16	KNNImputer	MinMax	ROS	SelectKBest	0.7713	0.0695
14	KNNImputer	MinMax	Nenhum	SelectKBest	0.7613	0.0272
8	Mediana	MinMax	ROS	SelectKBest	0.7581	0.0558
6	Mediana	MinMax	Nenhum	SelectKBest	0.7581	0.0273
15	KNNImputer	MinMax	ROS	Nenhuma	0.7572	0.0515
7	Mediana	MinMax	ROS	Nenhuma	0.7460	0.0689
4	Mediana	Standard	ROS	SelectKBest	0.7447	0.0589

3.2 Gráfico Comparativo

O gráfico abaixo ilustra o desempenho dos 16 pipelines com suas respectivas barras de erro (desvio padrão):



4. DISCUSSÃO E OBSERVAÇÕES

A análise dos resultados revela padrões claros sobre quais técnicas favorecem o desempenho do KNN para o dataset Hepatitis:

1. O Impacto da Imputação (KNNImputer vs. Mediana):

A técnica de imputação foi o fator mais determinante. O Experimento 9 utilizou KNNImputer.

- ➢ Observando o Top 2 (Exp 9 e Exp 13), ambos utilizam KNNImputer.

- ➢ Isso sugere que, para este dataset médico, os valores ausentes não são aleatórios simples. O KNNImputer consegue capturar correlações não lineares entre os sintomas para estimar o valor faltante com muito mais precisão do que a simples substituição pela Mediana global.

2. O Papel do Scaling (Standard vs. MinMax):

Embora ambos apareçam no topo, o StandardScaler obteve uma ligeira vantagem sobre o MinMaxScaler na melhor configuração. Isso indica que a distribuição dos dados pode se beneficiar mais da centralização na média e desvio padrão unitário, tornando o cálculo de distância do KNN menos sensível a extremos absolutos.

3. Balanceamento (Nenhum vs. Random Over-Sampling):

Um achado interessante foi que o Random Over-Sampling prejudicou o desempenho na maioria das combinações de topo.

- ➢ As 4 melhores configurações utilizam "Balanceamento: Nenhum".

- ➢ A introdução de dados sintéticos repetidos ROS pode ter causado overfitting ou ruído na fronteira de decisão do KNN, fazendo com que o modelo generalizasse pior nos dados de teste. Para este problema específico, preservar a distribuição original das classes foi mais benéfico.

4. Seleção de Atributos (Nenhuma vs. SelectKBest):

A redução de dimensionalidade para k=10 features consistentemente diminuiu o F1-Score.

- ➢ O melhor modelo utilizou todas as 19 features ("Nenhuma").

- ➢ Isso indica que, embora algumas features possam parecer menos relevantes individualmente, elas contribuem com informações úteis para a classificação em conjunto. Cortar quase metade das features (de 19 para 10) resultou em perda de informação valiosa para o KNN.

5. CONCLUSÃO

Principais Achados:

- ➢ A **melhor combinação** encontrada foi o Experimento 9: KNNImputer + StandardScaler + Sem Balanceamento + Sem Seleção.

- ➢ Esta configuração atingiu um F1-Score Médio de 0.8448, demonstrando a robustez do pipeline.

- ➢ Foi comprovado que técnicas de imputação mais sofisticadas como o KNNImputer superam imputações estatísticas simples como a Mediana em datasets complexos como o de Hepatite.

- ➢ Para este cenário, "menos não foi mais": reduzir features e introduzir dados sintéticos via Random Over-Sampling degradou a performance.

Lições Aprendidas: A atividade reforçou a importância de testar empiricamente combinações de pré-processamento. O que funciona na teoria nem sempre se traduz em melhoria prática para todos os datasets e algoritmos. O uso de pipelines com validação cruzada garantiu que essas conclusões fossem estatisticamente válidas e livres de vazamento de dados.

6. ANEXOS

Seguem links para melhor visualização das tabelas e dos gráficos:

- ➢ Link para imagem que mostra os 16 experimentos em forma de tabela;
- ➢ Link para tabela ordenada dos resultados pelo seu F1-Score e Desvio Padrão;
- ➢ Link para resultados dos gráficos de barras dos F1-score dos 16 pipelines;
- ➢ Link para o código no Drive para execução no VS Code;
- ➢ Link para pasta no Drive contendo todos os arquivos.