

Relatorio - Avaliacao de Modelos de Machine Learning

Datasets: Breast Cancer (Wisconsin Diagnostic) e Heart Failure Clinical Records

Objetivo. Implementar um conjunto de programas em Python que construa modelos de classificacao com Naive Bayes, Decision Tree, Random Forest e kNN, e calcule *accuracy* e *confusion matrix* para diferentes configuracoes de preparacao dos dados e diferentes percentagens de teste (15%, 30% e 50%).

Metodologia

Particao treino/teste. Para cada dataset foram efetuadas particoes estratificadas (preservando a distribuicao da classe) com **random_state=42** para garantir reprodutibilidade.

Variantes de pre-processamento. Foram avaliadas quatro configuracoes: (1) apenas codificacao do alvo (LabelEncoder para y); (2) + StandardScaler nas features numericas; (3) + OneHotEncoder para colunas categoricas (quando existirem); (4) OneHotEncoder + StandardScaler (escala apenas em features numericas).

Modelos. GaussianNB (Naive Bayes), DecisionTreeClassifier, RandomForestClassifier e KNeighborsClassifier. Para Random Forest foi usado `n_estimators=300` e `random_state=42`. Para kNN foi usado `k=5`.

Metricas. Alem de accuracy, foram calculadas precision, recall e f1 (todas com media *weighted*). A matriz de confusao foi gerada e guardada como imagem para cada execucao.

Resultados - Breast Cancer

Teste = 15%

Tabela: Accuracy (media por modelo)

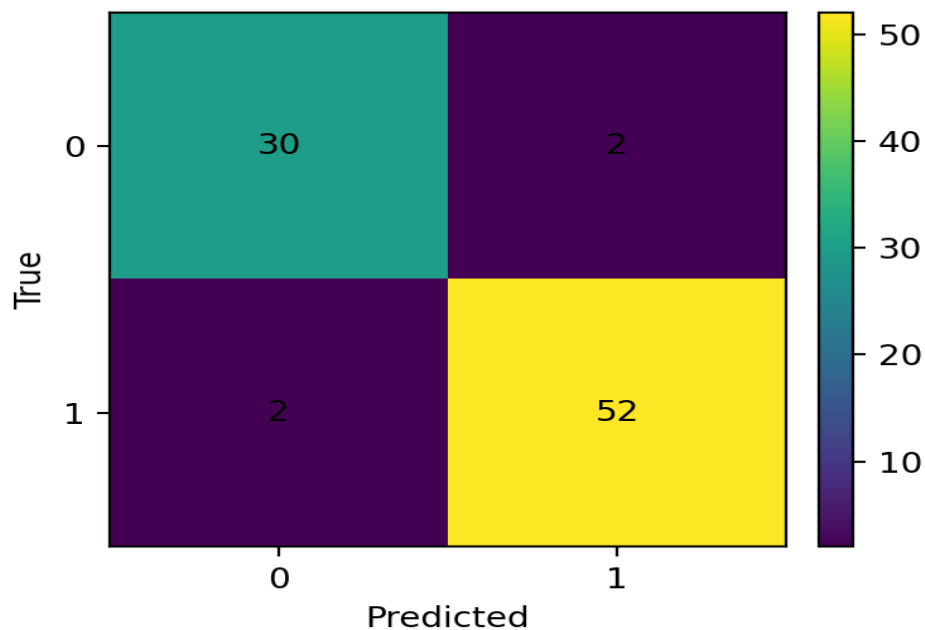
Preprocessing	Decision Tree	Naive Bayes	Random Forest	kNN
labelEncoder	0.942	0.942	0.953	0.930
labelEncoder+standardScaler	0.942	0.942	0.953	0.953
labelEncoder+oneHotEncoder	0.942	0.942	0.953	0.930
labelEncoder+oneHotEncoder+standardScaler	0.942	0.942	0.953	0.953

Tabela: Precision (weighted)

Preprocessing	Decision Tree	Naive Bayes	Random Forest	kNN
labelEncoder	0.945	0.942	0.953	0.930
labelEncoder+standardScaler	0.945	0.942	0.953	0.953
labelEncoder+oneHotEncoder	0.945	0.942	0.953	0.930
labelEncoder+oneHotEncoder+standardScaler	0.945	0.942	0.953	0.953

Melhor execucao (por accuracy): **Random Forest** com **labelEncoder**. Accuracy=0.953, Precision=0.953.

Breast Cancer | Random Forest | test=15%
labelEncoder



Teste = 30%

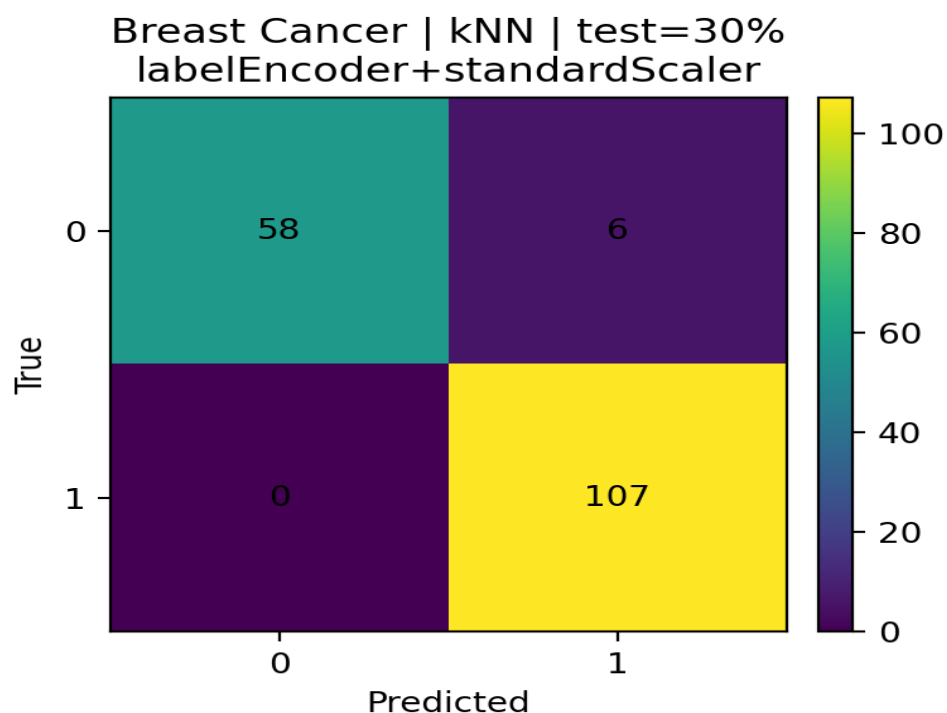
Tabela: Accuracy (media por modelo)

Preprocessing	Decision Tree	Naive Bayes	Random Forest	kNN
labelEncoder	0.918	0.947	0.942	0.930
labelEncoder+standardScaler	0.918	0.936	0.942	0.965
labelEncoder+oneHotEncoder	0.918	0.947	0.942	0.930
labelEncoder+oneHotEncoder+standardScaler	0.918	0.936	0.942	0.965

Tabela: Precision (weighted)

Preprocessing	Decision Tree	Naive Bayes	Random Forest	kNN
labelEncoder	0.918	0.948	0.941	0.930
labelEncoder+standardScaler	0.918	0.936	0.941	0.967
labelEncoder+oneHotEncoder	0.918	0.948	0.941	0.930
labelEncoder+oneHotEncoder+standardScaler	0.918	0.936	0.941	0.967

Melhor execucao (por accuracy): **kNN** com **labelEncoder+standardScaler**. Accuracy=0.965, Precision=0.967.



Teste = 50%

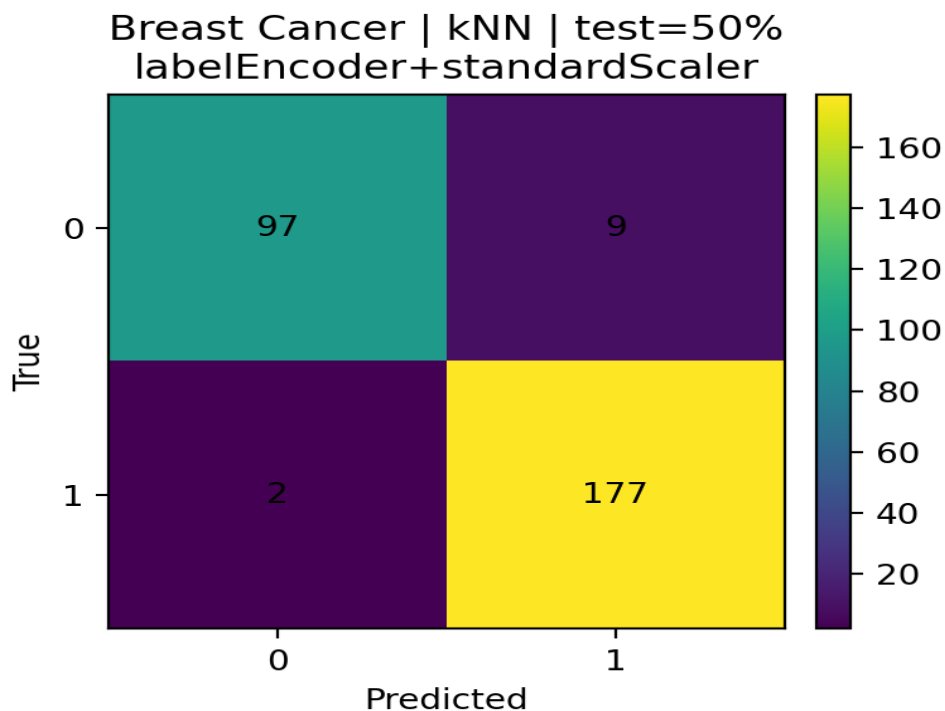
Tabela: Accuracy (media por modelo)

Preprocessing	Decision Tree	Naive Bayes	Random Forest	kNN
labelEncoder	0.912	0.944	0.947	0.930
labelEncoder+standardScaler	0.912	0.937	0.947	0.961
labelEncoder+oneHotEncoder	0.912	0.944	0.947	0.930
labelEncoder+oneHotEncoder+standardScaler	0.912	0.937	0.947	0.961

Tabela: Precision (weighted)

Preprocessing	Decision Tree	Naive Bayes	Random Forest	kNN
labelEncoder	0.913	0.945	0.947	0.930
labelEncoder+standardScaler	0.913	0.937	0.947	0.962
labelEncoder+oneHotEncoder	0.913	0.945	0.947	0.930
labelEncoder+oneHotEncoder+standardScaler	0.913	0.937	0.947	0.962

Melhor execucao (por accuracy): **kNN** com **labelEncoder+standardScaler**. Accuracy=0.961, Precision=0.962.



Analise critica (sintese)

O dataset Breast Cancer e relativamente equilibrado e contem apenas features numericas (30 variaveis). Nestas condicoes, algoritmos baseados em distancia (kNN) e modelos de conjunto (Random Forest) tendem a beneficiar da normalizacao (StandardScaler). Naive Bayes costuma ser um baseline forte, mas pode perder desempenho se as distribuicoes das features se afastarem da hipotese Gaussiana. Decision Tree pode ter variancia elevada, e por isso a Random Forest costuma melhorar a generalizacao. Quando o tamanho do teste aumenta (ex.: 50%), o treino fica menor e as diferencas entre modelos podem

aumentar; modelos mais robustos (RF) tendem a manter melhor desempenho.

Resultados - Heart Failure

Teste = 15%

Tabela: Accuracy (media por modelo)

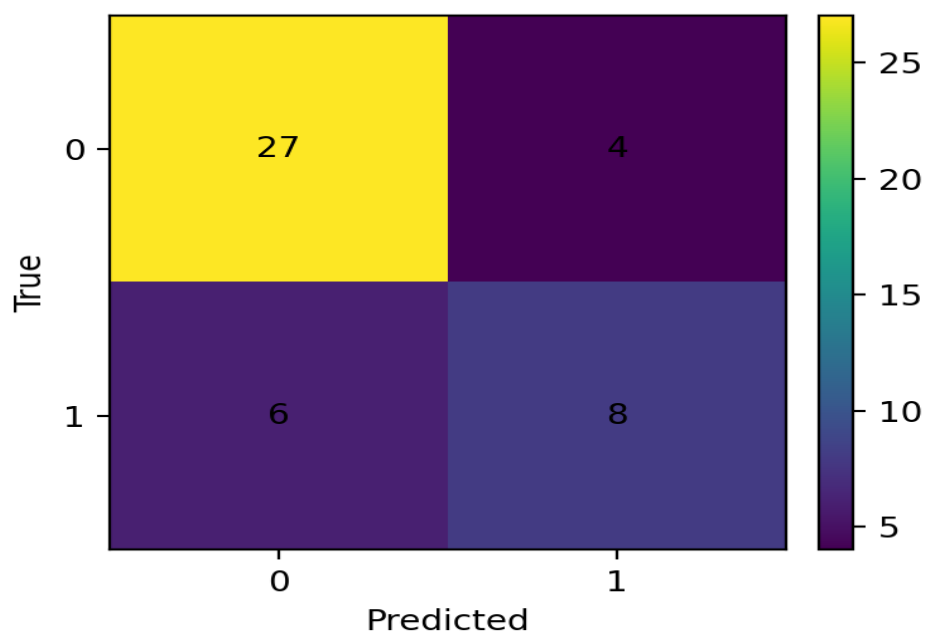
Preprocessing	Decision Tree	Naive Bayes	Random Forest	kNN
labelEncoder	0.733	0.733	0.778	0.644
labelEncoder+standardScaler	0.733	0.689	0.778	0.711
labelEncoder+oneHotEncoder	0.733	0.733	0.778	0.644
labelEncoder+oneHotEncoder+standardScaler	0.733	0.689	0.778	0.711

Tabela: Precision (weighted)

Preprocessing	Decision Tree	Naive Bayes	Random Forest	kNN
labelEncoder	0.718	0.716	0.771	0.581
labelEncoder+standardScaler	0.718	0.658	0.771	0.699
labelEncoder+oneHotEncoder	0.718	0.716	0.771	0.581
labelEncoder+oneHotEncoder+standardScaler	0.718	0.658	0.771	0.699

Melhor execucao (por accuracy): **Random Forest** com **labelEncoder**. Accuracy=0.778, Precision=0.771.

Heart Failure | Random Forest | test=15%
labelEncoder



Teste = 30%

Tabela: Accuracy (media por modelo)

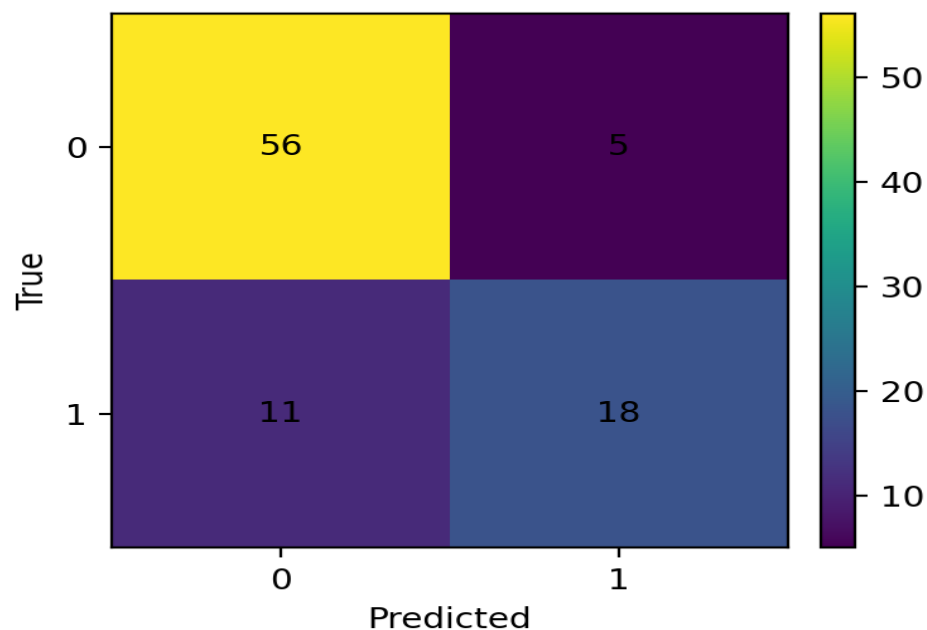
Preprocessing	Decision Tree	Naive Bayes	Random Forest	kNN
labelEncoder	0.789	0.756	0.822	0.567
labelEncoder+standardScaler	0.789	0.733	0.822	0.744
labelEncoder+oneHotEncoder	0.789	0.756	0.822	0.567
labelEncoder+oneHotEncoder+standardScaler	0.789	0.733	0.822	0.744

Tabela: Precision (weighted)

Preprocessing	Decision Tree	Naive Bayes	Random Forest	kNN
labelEncoder	0.784	0.747	0.819	0.514
labelEncoder+standardScaler	0.784	0.719	0.819	0.758
labelEncoder+oneHotEncoder	0.784	0.747	0.819	0.514
labelEncoder+oneHotEncoder+standardScaler	0.784	0.719	0.819	0.758

Melhor execucao (por accuracy): **Random Forest** com **labelEncoder**. Accuracy=0.822, Precision=0.819.

**Heart Failure | Random Forest | test=30%
labelEncoder**



Teste = 50%

Tabela: Accuracy (media por modelo)

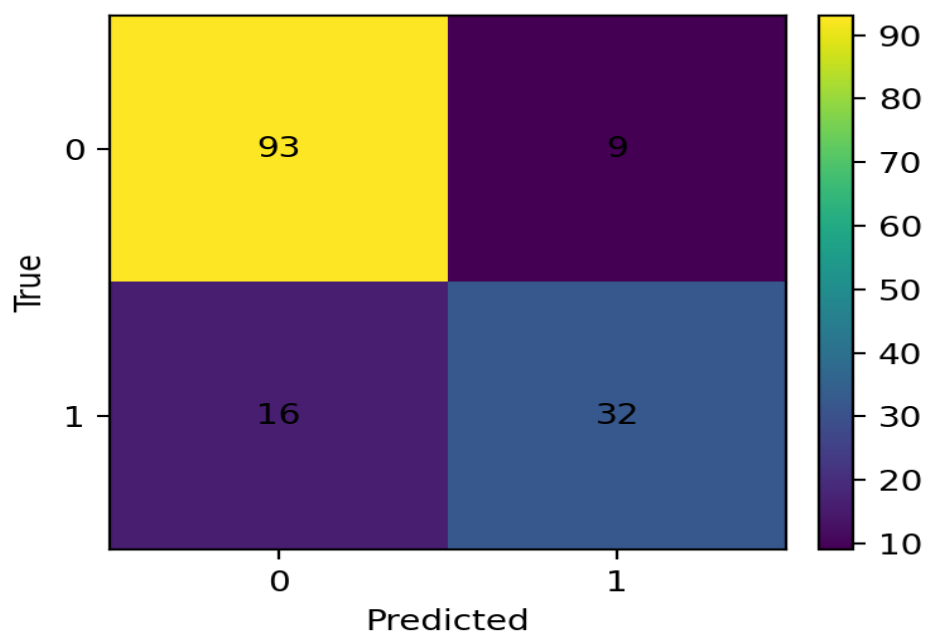
Preprocessing	Decision Tree	Naive Bayes	Random Forest	kNN
labelEncoder	0.740	0.753	0.833	0.587
labelEncoder+standardScaler	0.740	0.720	0.833	0.720
labelEncoder+oneHotEncoder	0.740	0.753	0.833	0.587
labelEncoder+oneHotEncoder+standardScaler	0.740	0.720	0.833	0.720

Tabela: Precision (weighted)

Preprocessing	Decision Tree	Naive Bayes	Random Forest	kNN
labelEncoder	0.736	0.742	0.830	0.504
labelEncoder+standardScaler	0.736	0.702	0.830	0.708
labelEncoder+oneHotEncoder	0.736	0.742	0.830	0.504
labelEncoder+oneHotEncoder+standardScaler	0.736	0.702	0.830	0.708

Melhor execucao (por accuracy): **Random Forest** com **labelEncoder**. Accuracy=0.833, Precision=0.830.

**Heart Failure | Random Forest | test=50%
labelEncoder**



Analise critica (sintese)

O dataset Heart Failure tem apenas 299 instancias e inclui variaveis binarias e continuas, com potencial desbalanceamento da classe (morte/nao-morte). Com poucos dados, o resultado fica mais sensivel a como o treino/teste e separado: ao aumentar o teste para 50%, o modelo treina com muito menos exemplos e a variancia aumenta. Decision Tree pode sobre-ajustar; Random Forest tende a ser mais estavel. kNN depende fortemente de escala, por isso a normalizacao costuma ser determinante. Como as features ja sao numericas/booleanas, o OneHotEncoder normalmente nao altera o desempenho (na pratica, nao ha colunas categoricas 'texto' a converter).

Reprodutibilidade

Para reproduzir localmente: (1) instalar dependencias: numpy, pandas, scikit-learn, matplotlib, reportlab; (2) executar **python run_experiments.py**; (3) executar **python make_report.py**. Os resultados e matrizes de confusao serao guardados na pasta **outputs/**.