

LAB #4: AVALIAÇÃO DE MODELOS DE IA

Objetivos

- Aplicar diferentes métricas de avaliação para classificação, regressão e clusterização
- Comparar o desempenho de múltiplos algoritmos
- Interpretar corretamente os resultados das métricas
- Identificar problemas como overfitting e underfitting

Parte 1: CLASSIFICAÇÃO

Dataset: Iris Dataset

Descrição: Classificação de espécies de flores Iris (Setosa, Versicolor, Virginica) baseado em medidas das sépalas e pétalas.

Tarefa 1.1: Implementação e Avaliação Básica

Implemente o classificador K-Nearest Neighbors (KNN) com $k=3$ e $k=5$

Questões:

- a) Divida o dataset em 70% treino e 30% teste. Para cada modelo, calcule:
 - Acurácia
 - Precisão (macro e micro)
 - Recall (macro e micro)
 - F1-score (macro e micro)
 - Matriz de confusão
- b) Qual modelo obteve melhor desempenho geral? Justifique sua resposta considerando múltiplas métricas.
- c) Existe alguma classe que é mais difícil de classificar? Como você identificou isso?

Tarefa 1.2: Validação Cruzada

- a) Implemente validação cruzada k-fold com $k=5$ para todos os modelos da Tarefa 1.1.
- b) Compare os resultados da validação cruzada com a divisão simples treino/teste. O que você observa?
- c) Calcule o desvio padrão da acurácia para cada modelo. O que isso indica sobre a estabilidade do modelo?

Tarefa 1.3: Curvas ROC e AUC

- a) Para o problema binário (Setosa vs. Não-Setosa), plote as curvas ROC para os três modelos.
- b) Calcule a área sob a curva (AUC) para cada modelo.
- c) Interprete os resultados: qual modelo tem melhor capacidade de discriminação?

Parte 2: REGRESSÃO

Dataset: Boston Housing (ou California Housing)

Descrição: Predição de preços de imóveis baseado em características socioeconômicas e geográficas.

Tarefa 2.1: Múltiplos Modelos de Regressão

Implemente o modelo. K-Nearest Neighbors Regressor ($k=3$ e $k=7$)

Questões:

- a) Divida o dataset em 70% treino e 30% teste. Para cada modelo, calcule:
 - Erro Médio Absoluto (MAE)
 - Erro Quadrático Médio (MSE)
 - Raiz do Erro Quadrático Médio (RMSE)
 - Coeficiente de Determinação (R^2)
 - Erro Percentual Absoluto Médio (MAPE)
- b) Crie um gráfico de dispersão comparando valores preditos vs. valores reais para cada modelo.
- c) Qual modelo apresentou melhor desempenho? Por quê?

Tarefa 2.2: Análise de Resíduos

- a) Para os três melhores modelos, plote o gráfico de resíduos (erro vs. valores preditos).
- b) Há algum padrão nos resíduos? O que isso indica sobre o modelo?
- c) Calcule e plote o histograma dos resíduos. Eles seguem uma distribuição aproximadamente normal?

Parte 3: CLUSTERIZAÇÃO

Dataset: Wine Dataset

Descrição: Análise química de vinhos de três cultivares diferentes na Itália.

Tarefa 3.1: K-Means

- a) Aplique o algoritmo K-Means com k variando de 2 a 10.
- b) Para cada valor de k, calcule:
 - Inércia (within-cluster sum of squares)
 - Silhouette Score
 - Davies-Bouldin Index
 - Calinski-Harabasz Index
- c) Plote o Método do Cotovelo (Elbow Method) usando a inércia.
- d) Determine o número ótimo de clusters considerando múltiplas métricas.

Tarefa 3.2: Comparação de Algoritmos

Implemente e compare:

- 1. K-Means (com k ótimo da Tarefa 3.1)
- 2. DBSCAN (teste diferentes valores de eps e min_samples)
- 3. Hierarchical Clustering (Agglomerative)

Questões:

- a) Calcule o Silhouette Score para cada algoritmo.
- b) Como os labels verdadeiros (cultivares) se comparam com os clusters encontrados?

Tarefa 3.3: Análise de Perfis dos Clusters

- a) Para a melhor solução de clusterização, calcule:
 - Média de cada feature por cluster
 - Desvio padrão de cada feature por cluster
- b) Crie um gráfico de radar (spider chart) mostrando o perfil médio de cada cluster.
- c) Descreva as características de cada cluster em termos das features originais.

Critérios de Avaliação

Aspectos Técnicos (60%)

- Implementação correta dos algoritmos
- Cálculo preciso das métricas
- Uso apropriado de validação cruzada
- Visualizações claras e informativas

Análise e Interpretação (30%)

- Justificativas bem fundamentadas
- Comparações significativas entre modelos
- Identificação de problemas (overfitting, underfitting, etc.)
- Recomendações baseadas em evidências

Apresentação (10%)

- Código bem documentado
- Relatório organizado e claro
- Gráficos com títulos, legendas e rótulos
- Conclusões objetivas

Recursos e Ferramentas Sugeridas

Bibliotecas Python:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.datasets import load_iris, load_wine, load_breast_cancer, fetch_california_housing
from sklearn.model_selection import train_test_split, cross_val_score, KFold
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score, f1_score,
    confusion_matrix, classification_report,
    roc_curve, roc_auc_score, auc,
    mean_absolute_error, mean_squared_error, r2_score,
    silhouette_score, davies_bouldin_score, calinski_harabasz_score,
    adjusted_rand_score
)

from sklearn.neighbors import KNeighborsClassifier, KNeighborsRegressor
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.cluster import KMeans, DBSCAN, AgglomerativeClustering
```

Datasets Alternativos:

- **Classificação:** Titanic, MNIST (dígitos), Wine Quality
- **Regressão:** Diabetes, Auto MPG, Bike Sharing
- **Clusterização:** Mall Customers, Wholesale Customers

Entregas Esperadas

1. Notebook Jupyter com:

- Código completo e executável
- Comentários explicativos
- Todas as saídas visíveis

Dicas para o Sucesso

1. **Comece simples:** Implemente um modelo básico primeiro, depois adicione complexidade
2. **Documente tudo:** Explique o que cada célula de código faz
3. **Compare sempre:** Um resultado isolado tem pouco valor; sempre compare múltiplos modelos
4. **Visualize:** Um bom gráfico vale mais que mil números
5. **Contextualize:** Relacione os resultados com o problema real
6. **Seja crítico:** Questione os resultados, busque entender o “porquê”

Prazo e Formato de Entrega

Prazo: 23/11/2025

Formato: Submissão via Classroom - Arquivo .ipynb (Jupyter Notebook)

Trabalho: Individual

Referências

- Scikit-learn Documentation: <https://scikit-learn.org/>
- Géron, A. “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow”
- James, G. et al. “An Introduction to Statistical Learning”
- UCI Machine Learning Repository:
<https://archive.ics.uci.edu/ml/>

Bom trabalho! Lembre-se: a avaliação de modelos é tão importante quanto a construção deles.