



Engenharia de Computação
Inteligência Artificial
11 - Atividade Integradora
Comparação entre DT, KNN e SVM

Organização da atividade

- Retomada: três estratégias de classificação – Árvore de Decisão, KNN e SVM.
- Matriz de confusão: verdadeiro/falso positivo e verdadeiro/falso negativo.
- Métricas: acurácia, precisão, *recall* e F1.
- Experimento controlado: mesma base, mesma divisão treino/teste e três classificadores.
- Comparação dos resultados e escolha justificada de um modelo.
- Após a consolidação: apresentação do Trabalho 2.

Fio condutor

Se três modelos resolvem o mesmo problema, o que significa dizer que um deles é “melhor”?

Retomada: três formas de classificar

Árvore de Decisão

Aprende regras hierárquicas que particionam o espaço de atributos.

KNN

Classifica pela proximidade entre o novo exemplo e exemplos conhecidos.

SVM

Aprende uma fronteira de
decisão baseada em
margem e vetores de
suporte.

Mesma tarefa, hipóteses diferentes

Os três podem receber o mesmo conjunto de atributos e produzir a mesma classe de saída, mas constroem a decisão de formas muito diferentes.

Acurácia é suficiente?

Considere 100 exemplos, dos quais apenas 5 pertencem à classe positiva.

Um classificador responde **negativo para todos**.

Resultado

Ele acerta 95 dos 100 exemplos: **95% de acurácia**.

Pergunta

Podemos concluir que esse é um bom classificador se a classe positiva for justamente a classe de maior interesse?

Matriz de confusão

Para classificação binária, as previsões podem ser organizadas em quatro grupos:

	Predito negativo	Predito positivo
Real negativo	TN	FP
Real positivo	FN	TP

- **TP**: positivo corretamente identificado;
- **TN**: negativo corretamente identificado;
- **FP**: negativo classificado incorretamente como positivo;
- **FN**: positivo classificado incorretamente como negativo.

Observação

Antes de interpretar precisão e *recall*, precisamos saber qual classe foi definida como **positiva**.

Quatro métricas para comparar os modelos

Acurácia

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Fração total de previsões corretas.

Recall

$$\frac{TP}{TP + FN}$$

Dos positivos reais, quantos foram encontrados?

Precisão

$$\frac{TP}{TP + FP}$$

Dos exemplos previstos como positivos, quantos realmente eram positivos?

F1

$$2 \frac{Precision \cdot Recall}{Precision + Recall}$$

Média harmônica entre precisão e *recall*.

Precisão ou recall: qual é mais importante?

Precisão alta

Quando o modelo diz “positivo”, queremos que ele esteja correto.

Penaliza especialmente **falsos positivos**.

Recall alto

Queremos encontrar a maior quantidade possível dos positivos reais.

Penaliza especialmente **falsos negativos**.

Decisão depende do problema

Em qual aplicação seria mais grave deixar um positivo real escapar? E em qual aplicação um falso alarme teria custo elevado?

Base da atividade: Breast Cancer Wisconsin

Nesta atividade utilizaremos a base disponibilizada diretamente pelo *Scikit-learn*:

- problema de **classificação binária**;
- atributos numéricos derivados de características de núcleos celulares;
- duas classes: *malignant* e *benign*;
- não será necessário tratamento de categorias ou campos ausentes nesta atividade.

```
from sklearn.datasets import load_breast_cancer

data = load_breast_cancer()
X = data.data
```

Escolha didática

A preparação de dados já foi trabalhada na aula de KNN. Hoje o foco é comparar classificadores em um experimento controlado.

Qual será a classe positiva?

Na base original do *Scikit-learn*:

- 0 representa *malignant*;
- 1 representa *benign*.

Para esta atividade, vamos definir **maligno como a classe positiva**:

```
y = (data.target == 0).astype(int)
```

Agora:

- 1 = maligno – positivo;
- 0 = benigno – negativo.

Interpretação

Nesse contexto, qual erro corresponde a um caso maligno que o modelo classificou como benigno?

Protocolo do experimento

Para que a comparação seja justa, os três modelos devem usar o mesmo protocolo:

- 1 mesma base de dados;
- 2 mesma definição de x e y ;
- 3 mesma divisão entre treino e teste;
- 4 mesma semente aleatória;
- 5 nenhuma escolha de hiperparâmetro baseada no conjunto de teste;
- 6 mesmas métricas de avaliação.

Controle experimental

Se alterarmos simultaneamente o algoritmo e os dados de teste, deixamos de saber qual mudança explica a diferença observada.

Divisão treino/teste

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.25,
    stratify=y,
    random_state=42
)
```

- 75% dos exemplos: treinamento;
- 25% dos exemplos: teste;
- `stratify=y`: preserva aproximadamente a proporção das classes;
- `random_state=42`: torna a divisão reproduzível.

Pré-processamento: igual para todos?

DT

Não depende de distância entre exemplos. Nesta atividade, não faremos padronização.

KNN

Depende diretamente das distâncias. Usaremos `StandardScaler`.

SVM

A geometria da margem/kernel é sensível à escala. Usaremos `StandardScaler`.

Pergunta

Uma comparação justa significa obrigatoriamente aplicar exatamente o mesmo pré-processamento a todos os algoritmos?

Configurações da atividade

Não faremos *tuning* hoje. Utilizaremos configurações fixas e razoáveis:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler

model_dt = DecisionTreeClassifier(
    max_depth=4, random_state=42
)

model_knn = make_pipeline(
    StandardScaler(),
    KNeighborsClassifier(n_neighbors=5)
)

model_svm = make_pipeline(
    StandardScaler(),
    SVC(kernel="rbf", C=1.0, gamma="scale")
)
```

Atividade integradora – 25 minutos

Em duplas ou trios

Treine os três classificadores usando exatamente a mesma divisão dos dados e compare seus resultados.

- 1 treine DT, KNN e SVM;
- 2 faça previsões apenas sobre `X_test`;
- 3 calcule acurácia, precisão, *recall* e F1;
- 4 gere a matriz de confusão de cada modelo;
- 5 registre os resultados em uma tabela;
- 6 escolha um modelo e justifique sua decisão.

Observação

Não altere `max_depth`, `k`, `C` ou `gamma` para tentar “ganhar” a comparação. O objetivo é avaliar, não otimizar.

Função para avaliar os modelos

```
from sklearn.metrics import (
    accuracy_score,
    precision_score,
    recall_score,
    f1_score,
    confusion_matrix
)

def avaliar(nome, y_true, y_pred):
    print(f"\n{name}")
    print("Accuracy :", accuracy_score(y_true, y_pred))
    print("Precision:", precision_score(y_true, y_pred))
    print("Recall   :", recall_score(y_true, y_pred))
    print("F1       :", f1_score(y_true, y_pred))
    print(confusion_matrix(y_true, y_pred))
```

Antes de executar

Como a definição de maligno como classe positiva afeta a interpretação de *precision* e *recall*?

Tabela de resultados

Preencha após executar os três modelos:

Modelo	Acurácia	Precisão	Recall	F1
Árvore de Decisão				
KNN				
SVM				

Não pare na maior acurácia

Algum modelo apresenta uma troca relevante entre precisão e *recall*?

Questões para análise

Discussão final – 5 minutos

Responda com base nos resultados obtidos, e não apenas na definição teórica dos algoritmos.

- 1 Qual modelo obteve maior acurácia?
- 2 Qual obteve maior *recall* para casos malignos?
- 3 Qual obteve maior F1?
- 4 Quantos falsos negativos cada modelo produziu?
- 5 Se um falso negativo for considerado o erro mais grave, qual modelo você escolheria?
- 6 É possível afirmar, a partir deste único experimento, que esse algoritmo é sempre superior aos demais?

Resposta de referência – métricas

Com a divisão e configurações usadas nesta atividade, uma execução reproduzível produz aproximadamente:

Modelo	Acurácia	Precisão	Recall	F1
Árvore de Decisão	0,895	0,975	0,736	0,839
KNN	0,958	0,980	0,906	0,941
SVM	0,972	1,000	0,925	0,961

Observação

Esses valores são consequência desta base, desta divisão e destas configurações. Eles não constituem um ranking universal entre os algoritmos.

Resposta de referência – matrizes de confusão

Considerando **maligno = positivo**:

DT

$$\begin{bmatrix} 89 & 1 \\ 14 & 39 \end{bmatrix}$$

FN = 14

KNN

$$\begin{bmatrix} 89 & 1 \\ 5 & 48 \end{bmatrix}$$

FN = 5

SVM

$$\begin{bmatrix} 90 & 0 \\ 4 & 49 \end{bmatrix}$$

FN = 4

Leitura da matriz

No padrão do *Scikit-learn*: $[[TN, FP], [FN, TP]]$ para as classes ordenadas como 0 e 1.

O que podemos concluir?

- neste experimento, a SVM apresentou os maiores valores das quatro métricas;
- o KNN também teve desempenho elevado;
- a Árvore produziu mais falsos negativos com a configuração fixada;
- a conclusão depende da base, da divisão, do pré-processamento e dos hiperparâmetros;
- métricas diferentes respondem a perguntas diferentes sobre os erros do modelo.

Conclusão crítica

Se repetirmos o experimento com outra base, outra divisão ou outros hiperparâmetros, somos obrigados a obter a mesma ordem entre os modelos?

Erros conceituais frequentes

- comparar modelos treinados/testados em divisões diferentes;
- ajustar hiperparâmetros olhando continuamente o conjunto de teste;
- comparar KNN/SVM sem considerar a escala dos atributos;
- interpretar precisão ou *recall* sem saber qual é a classe positiva;
- concluir que o algoritmo de maior acurácia é automaticamente o melhor para qualquer aplicação;
- tratar um resultado obtido em uma única base como evidência universal de superioridade.

Síntese da consolidação

Problema → dados → pré-processamento →
modelo → previsão → avaliação → decisão

- DT, KNN e SVM representam maneiras diferentes de construir um classificador;
- uma comparação exige um protocolo experimental comum;
- a matriz de confusão revela os tipos de erro;
- acurácia, precisão, *recall* e F1 complementam a análise;
- a escolha final depende também do custo dos erros no problema real.

Na sequência: Trabalho 2

Da comparação controlada de três classificadores para um projeto maior de Inteligência Artificial.

Apresentação do Trabalho 2

Após esta atividade, serão apresentados o problema, as etapas, os métodos que serão incorporados progressivamente, os entregáveis e os critérios de avaliação do Trabalho 2.

Leve esta ideia para o trabalho

Não basta executar algoritmos: será necessário justificar representação, tratamento dos dados, escolhas metodológicas e interpretação dos resultados.

Dúvidas



Dúvida é o começo da sabedoria.