

Classificação

Módulo 07 - Sistemas de Informação Inteligentes

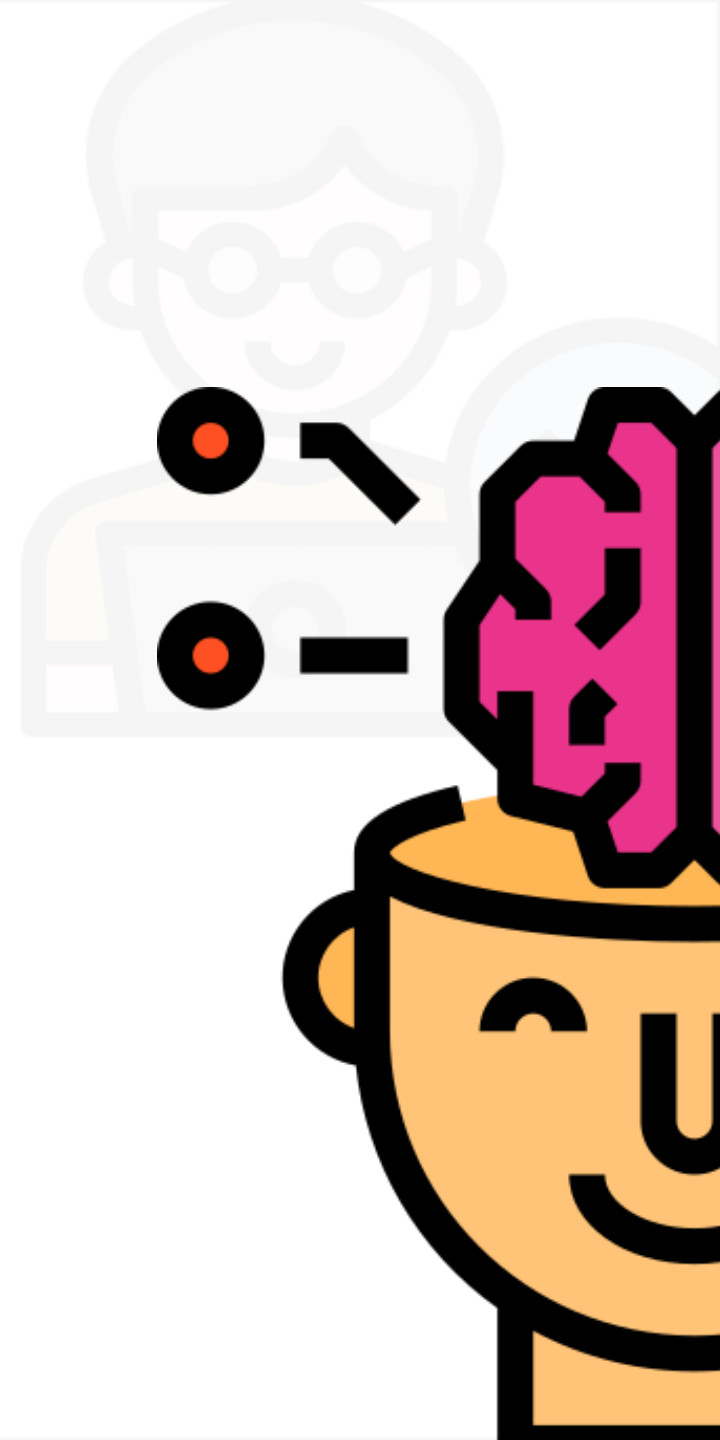
ANTÔNIO DAVID VINISKI

antonio.david@pucpr.br

PUCPR

Tópicos

- Avaliação dos modelos de Classificação
 - Acurácia.
 - Precisão.
 - Recall.
 - F1-score.
- Algoritmos de Classificação
 - Regressão Logística.
 - K vizinhos mais próximos.
 - Árvores de Decisão.
 - Redes Bayesianas.
 - Máquinas de vetores e suporte para regressão.
 - Redes Neurais





Matriz de Confusão

Matriz de Confusão Binária

		Valor real	
		Sim	Não
Valor predito	Sim	TP	FP
	Não	FN	TN

TP: Verdadeiros Positivos

FP: Falsos Positivos

FN: Falsos Negativos

TN: Verdadeiros Negativos



Acurácia

$$\textit{Acurácia} = \frac{TP + TN}{TP + TN + FP + FM}$$

- Nos diz quantos de nossos exemplos foram de fato classificados corretamente, independente da classe.
- Uma das maiores desvantagens é que em alguns problemas a acurácia pode ser elevada mas, ainda assim, o modelo pode ter uma performance inadequada.
 - Não deve ser utilizada em problemas onde o número de exemplos por classe é desbalanceado.



Precisão

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



- A precisão dá uma ênfase maior para os erros por falso positivo.
 - É uma expressão matemática para pergunta: **Dos exemplos classificados como positivos, quantos realmente são positivos?**

Revocação (Recall)

- *Recall* em inglês e também conhecida como **sensibilidade** ou taxa de verdadeiro positivo (TPR).

$$\textit{Recall} = \frac{TP}{TP + FN}$$

- Busca responder a seguinte pergunta: De todos os exemplos que são positivos, quantos foram classificados corretamente como positivos?



F1-score

- Também conhecida como *F-measure*, considera tanto a precisão quanto a revocação, sendo definida pela média harmônica entre as duas:

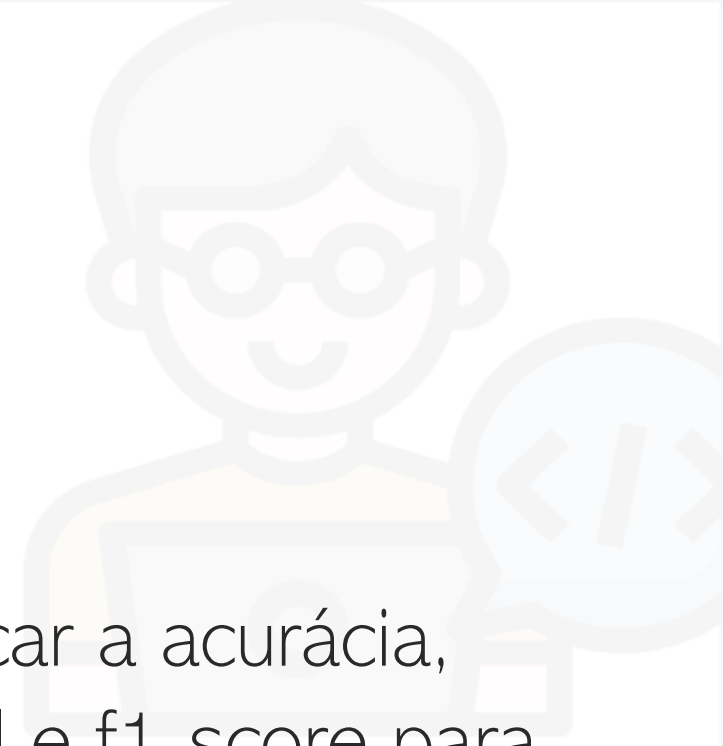
$$F_1 = 2 * \frac{precisao * recall}{precisao + recall}$$

- Para que o F1-score seja alto, tanto a precisão como a revocação também devem ser altas.
- Tende a ser um resumo melhor da qualidade do modelo.
- Uma desvantagem é que a F1 acaba sendo menos interpretável que a acurácia.

Matriz de Confusão Múltiplas Classes

		Valor real		
		Gato	Peixe	Cão
Valor predito	Gato	4	6	3
	Peixe	1	2	0
	Cão	1	2	6

- Como identificar a acurácia, precisão, recall e f1-score para um problema com múltiplas classes?



Precisão – múltiplas classes

		Valor real		
		Gato	Peixe	Cão
Valor predito	Gato	4	6	3
	Peixe	1	2	0
	Cão	1	2	6

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

- Similar a um problema binário, podemos definir a previsão do modelo de classificação para cada uma das classes.
- Qual seria a precisão da classe gato?

$$P_{gato} = \frac{4}{4 + 6 + 3} = \frac{4}{13} = 0,308$$

Recall – múltiplas classes

		Valor real		
		Gato	Peixe	Cão
Valor predito	Gato	4	6	3
	Peixe	1	2	0
	Cão	1	2	6

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

- No caso do recall, o valor corresponderia ao número de fotos de gatos preditas corretamente.
- Qual seria o recall da classe gato?

$$R_{gato} = \frac{4}{4 + 1 + 1} = \frac{4}{6} = 0,667$$

F1-score – múltiplas classes

		Valor real		
		Gato	Peixe	Cão
Valor predito	Gato	4	6	3
	Peixe	1	2	0
	Cão	1	2	6

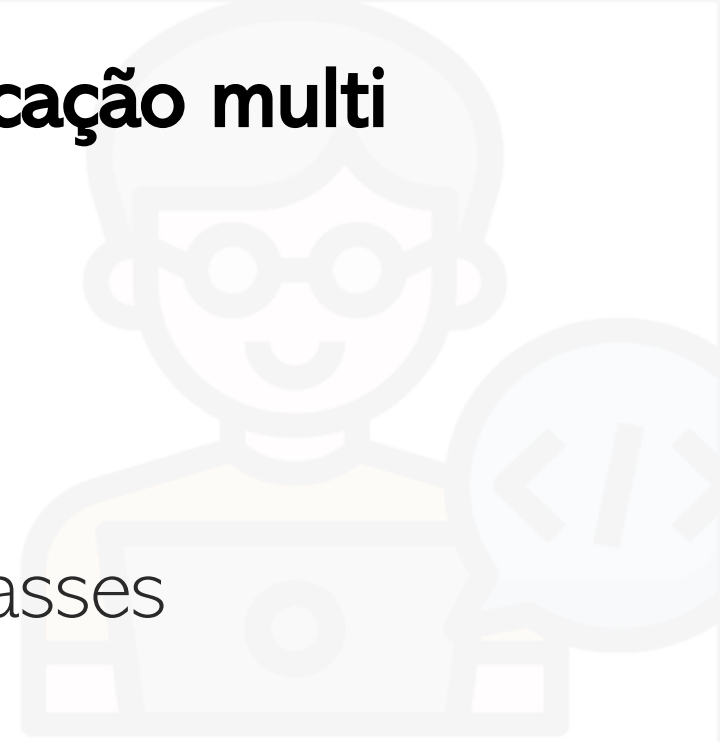
$$F_1 = 2 * \frac{precisao * recall}{precisao + recall}$$

- Utiliza-se a precisão e o recall calculado para a respectiva classe.
- Qual seria o f1-score da classe gato?

$$F1_{gato} = 2 * \frac{0,308 * 0,667}{0,308 + 0,667} = 0,421$$


E como calcular o resultado geral da classificação multi classes?

- **Micro-avg:** Utilizando os dados de todas as classes simultaneamente.
- **Macro avg:** Utilizando a media dos valores da métrica para cada classe.
- **Weighted avg:** Utilizando a media ponderada para cada classe.



Micro-avg precision

	TP	FP	FN
Gato	4	9	2
Peixe	2	1	8
Cão	6	3	3
Total	12	13	13

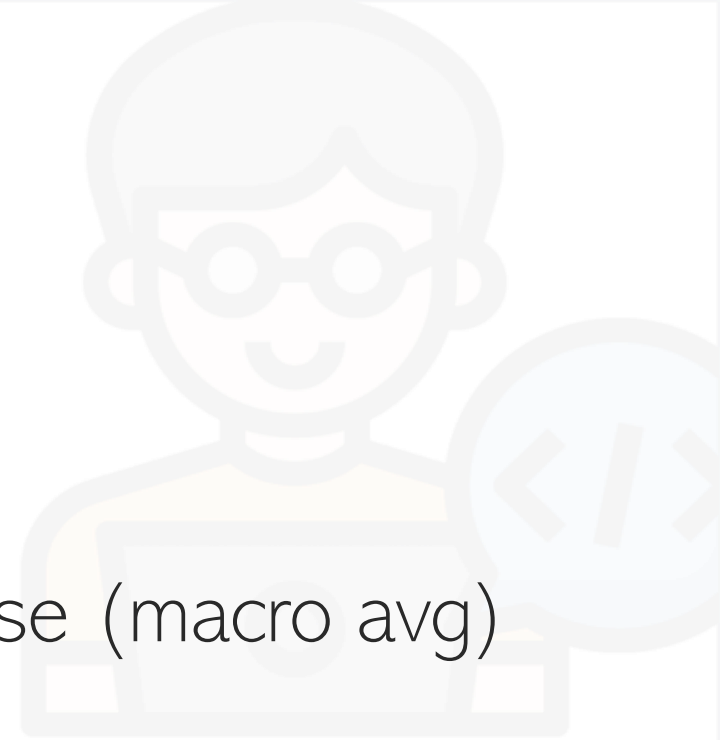

$$\text{micro avg precision} = \frac{TP_{total}}{TP_{total} + FP_{total}}$$

$$\text{micro avg precision} = \frac{12}{12+13} = 0,48$$

Macro-avg

- Utilizando a media dos valores para cada classe (macro avg)

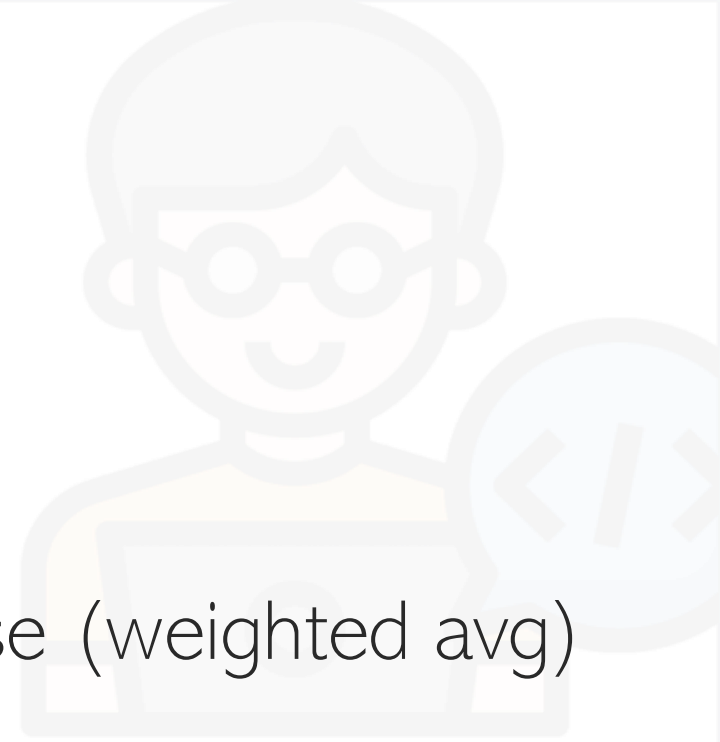
$$\mathbf{macro\ avg} = \frac{P_{gato} + P_{peixe} + P_{cao}}{N_{classes}} = ?$$



Weighted-avg

- Utilizando a média ponderada para cada classe (weighted avg)

$$\textit{weighted avg} = \frac{P_{gato} * N_{gato} + P_{peixe} * N_{peixe} + P_{cao} * N_{cao}}{N_{total}} = ?$$



Exercício 1

- Calcular a média micro, macro e ponderada para cada uma das métricas de avaliação apresentadas, considerando o problema de classificação dos animais.

		Valor real		
		Gato	Peixe	Cão
Valor predito	Gato	4	6	3
	Peixe	1	2	0
	Cão	1	2	6

	TP	FP	FN
Gato	4	9	2
Peixe	2	1	8
Cão	6	3	3
Total	12	13	13

Sklearn métricas classificação

- Considerando o exemplo anterior, vamos modelar o problema.

```
# Constants
C="Cat"
F="Fish"
D="Dog"

# True values
y_true = [C,C,C,C,C,C, F,F,F,F,F,F,F,F,F, D,D,D,D,D,D,D,D,D]
# Predicted values
y_pred = [C,C,C,C,D,F, C,C,C,C,C,C,D,D,F,F, C,C,C,D,D,D,D,D,D]
```

Sklearn métricas para classificação

```
from sklearn.metrics import confusion_matrix, classification_report
# Print the confusion matrix
print(confusion_matrix(y_true, y_pred))

# Print the precision and recall, among other metrics
print(classification_report(y_true, y_pred, digits=3))

from sklearn.metrics import precision_score, f1_score, recall_score
print("Precision - micro",precision_score(y_true,y_pred,average="micro"))
print("Recall - micro",recall_score(y_true,y_pred,average="micro"))
print("F1-score - micro",f1_score(y_true,y_pred,average="micro"))

print("Precision - macro",precision_score(y_true,y_pred,average="macro"))
print("Recall - macro",recall_score(y_true,y_pred,average="macro"))
print("F1-score - macro",f1_score(y_true,y_pred,average="macro"))

print("Precision - weighted",precision_score(y_true,y_pred,average="weighted"))
print("Recall - weighted",recall_score(y_true,y_pred,average="weighted"))
print("F1-score - weighted",f1_score(y_true,y_pred,average="weighted"))
```

Classificação Scikit-Learn

○ Principais algoritmos

```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
```

<https://tatianaesc.medium.com/implementando-um-modelo-de-classifica%C3%A7%C3%A3o-no-scikit-learn-6206d684b377>

