



Métricas de avaliação II: Recall e Precision.

☰ Ciclo	Ciclo 02: Aprendizado supervisionado - Classificação
# Aula	15
🕒 Created	@January 5, 2023 4:41 PM
☑ Done	☑
☑ Ready	☑

Objetivo da Aula:

- ☐ Precision & Recall
- ☐ Exemplos práticos
- ☐ Métricas na prática
- ☐ Resumo
- ☐ Próxima Aula

Conteúdo:

▼ 1. Precision & Recall

Vamos imaginar que você reservou o sábado inteiro para ir ao pesqueiro. Você leva no seu kit pesca duas varas de pesca. A primeira é mais comprida e portanto tem um alcance maior, enquanto a segunda é mais curta, tendo um alcance menor.

Quando você chega no lago, você joga a vara 160 vezes no lago. Você consegue pescar 70 peixes. Portanto, a sua precisão na pesca é $70/160$ igual a 44%.

No meio do dia, você vai até o restaurante do pesqueiro e começa a conversar com o dono. Ele revela para você, que no tanque onde você está pescando, existem 100 peixes no total. Logo, você

pescou 70 peixes do total de 100 peixes disponível no lago. Portanto, a sua recall é de 70%.

Após o almoço, você decide usar a segunda vara de pesca que é um pouco menor do que a primeira e portanto tem um alcance menor do lago.

Você joga a segunda vara 160 vezes no lago. Você consegue pescar 30 peixes dos 30 que sobraram. Portanto, sua precisão é $30 / 160$ igual a 19%, enquanto o Recall é $30/30$ igual a 100%.

Abaixo estão os resultados da sua pescaria:

	Pescados	Total de peixes	Precision	Recall
Manhã	70	100	44%	70%
Tarde	30	30	19%	100%

▼ 1.2 Fórmula da Precisão

▼ Matriz de confusão

		PREDICTED	
		POSITIVE	NEGATIVE
ACTUAL	POSITIVE	TRUE POSITIVES	FALSE NEGATIVES
	NEGATIVE	FALSE POSITIVES	TRUE NEGATIVES

▼ Precisão

$$Precisão = \frac{TP}{TP + FP}$$

▼ 1.3 Fórmula do Recall

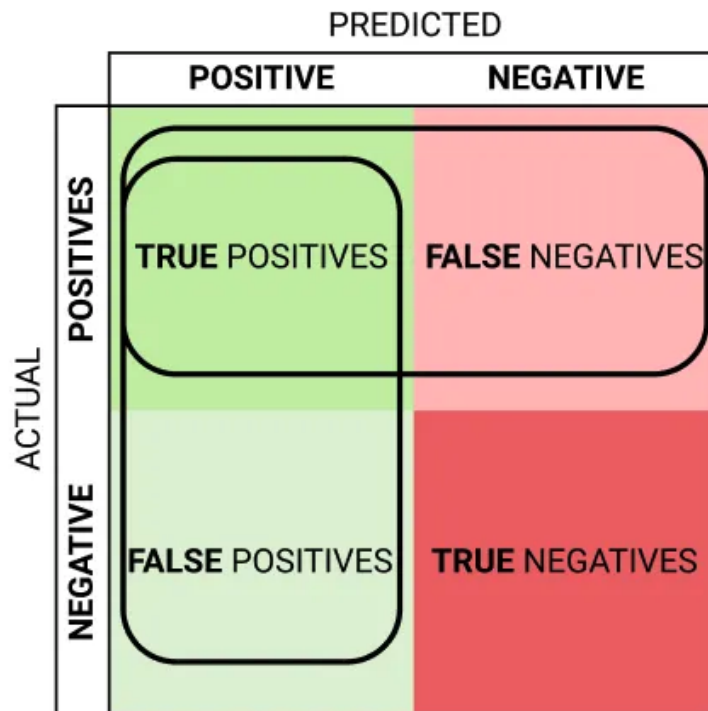
▼ Matriz de confusão

		PREDICTED	
		POSITIVE	NEGATIVE
ACTUAL	POSITIVE	TRUE POSITIVES	FALSE NEGATIVES
	NEGATIVE	FALSE POSITIVES	TRUE NEGATIVES

▼ Recall

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

▼ 1.4 Resumo



▼ Precisão

$$Precisão = \frac{TP}{TP + FP}$$

▼ Recall

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

▼ 2. Exemplos práticos

▼ Exemplo 01:

Você está tentando prever o Churn do cliente, ou seja, o momento em que ele vai parar de usar o seu produto. Isso pode acontecer quando ele não renova o contrato por mais um período de tempo, quando ele para de acessar o site e etc. Os resultados do treinamento do algoritmo são mostrados abaixo:

		Previsão	
		Churn	Not Churn
n = 400			
Real	Churn	300	35

No Churn 15 50

▼ Qual seria a **métricas mais adequada** para medir a performance desse classificador?

Acurácia: $(300 + 50) / (300 + 50 + 35 + 15) = 87,5\%$

Precisão: $300 / (300 + 15) = 95\%$

Recall: $300 / (300 + 35) = 89\%$

▼ **Exemplo 02:**

Você foi contratado como Cientista de Dados para classificar se um produto será devolvido pela cliente ou não. Os resultados do treinamento do algoritmo são mostrados abaixo:

		Previsão	
		Devolução	Sem devolução
Real	Devolução	20	80
	Sem devolução	60	50

▼ Qual seria a **métricas mais adequada** para medir a performance desse classificador?

Acurácia: $(20 + 50) / (20 + 80 + 60 + 50) = 33,3\%$

Precisão: $20 / (20 + 60) = 25\%$

Recall: $20 / (20 + 80) = 20\%$

▼ **3. Métricas na prática**

```
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics as mt

# 1.0 Load dataset
df = pd.read_csv( '../dataset/train.csv' )

# 2.0 Seleção de Features
features = ['idade', 'saldo_atual', 'divida_atual', 'renda_anual', 'valor_em_investimentos',
            'taxa_utilizacao_credito', 'num_emprestimos', 'num_contas_bancarias', 'num_cartoes_credito',
            'dias_atraso_dt_venc', 'num_pgtos_atrasados', 'num_consultas_credito', 'taxa_juros']

label = ['limite_adicional']

x_train = df.loc[:, features]
y_train = df.loc[:, label].values.ravel()
```

```
# 3.0 Treinamento do KNN
knn_classifier = KNeighborsClassifier( n_neighbors = 8 )
knn_classifier.fit( x_train, y_train )

y_pred = knn_classifier.predict( x_train )
# 4.0 Performance

# 4.1 Confusion Matrix
mt.confusion_matrix( y_train, y_pred )

# 4.2 Accuracy
mt.accuracy_score( y_train, y_pred )

# 4.3 Precisão
mt.precision_score( y_train, y_pred, average="binary", pos_label='Conceder' )

# 4.4 Recall
mt.recall_score( y_train, y_pred, average='binary', pos_label='Conceder' )
```

▼ 4. Resumo

1. Precisão é o número de acertos a cada tentativa
2. Recall é de todas as possibilidades de acerto, quantas você conseguiu acertar.

▼ 5. Próxima aula

Exercícios