



Métricas de avaliação I: Curva Precision x Recall

☰ Ciclo	Ciclo 06: Algoritmos baseado em árvores
# Aula	44
🕒 Created	@March 9, 2023 10:11 AM
☑ Done	<input type="checkbox"/>
☑ Ready	<input checked="" type="checkbox"/>

Objetivo da Aula:

- ☐ Revendo a Precision & Recall
- ☐ Resumo
- ☐ Próxima aula

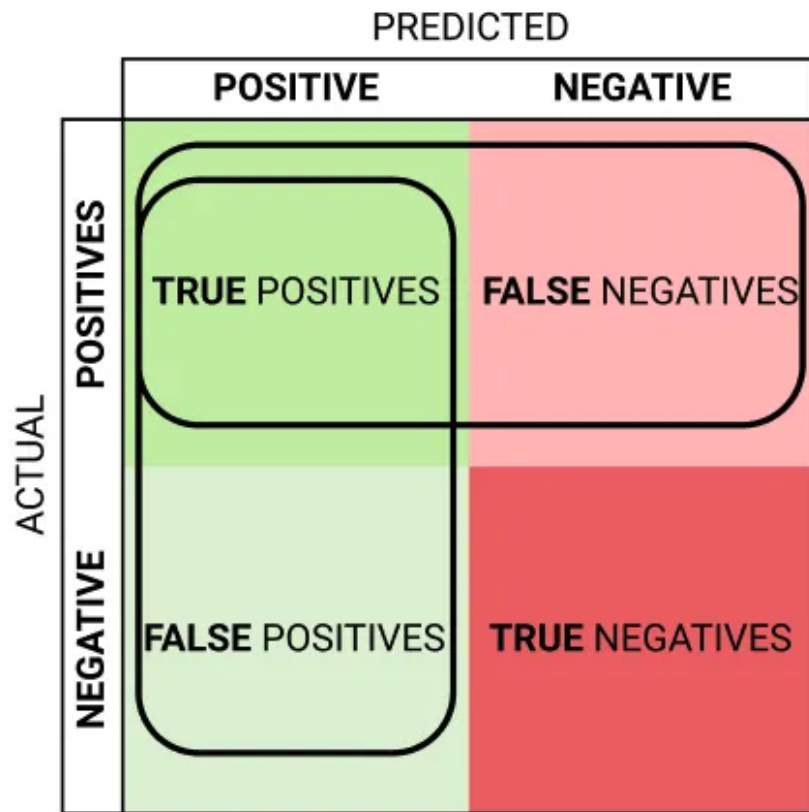
Conteúdo:

▼ 1. Revendo a Precision & Recall

Vamos imaginar que você reservou o sábado inteiro para ir ao pesqueiro. Abaixo estão os resultados da sua pescaria:

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

▼ 1.1 Matriz de confusão



▼ Precisão

$$Precisão = \frac{TP}{TP + FP}$$

▼ Recall

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

▼ 2. Precision/Recall TradeOff

Em alguns problemas de negócio, a métrica mais importante é a precisão, enquanto outros a métrica mais importante é recall.

Por exemplo, a empresa onde você trabalha desenvolve uma plataforma de hospedagem de vídeos de produtores de conteúdo com foco em crianças. Sua

principal responsabilidade como Cientista de Dados da empresa, é detectar os vídeos seguros para crianças assistirem.

Nesse cenário, seria preferível um classificador que rejeitasse muitos bons vídeos classificando-os como “inapropriados”, mas conseguisse classificar todos os vídeos realmente inapropriados para menores como “inapropriados” (alta precisão).

Por outro lado, imagine que você trabalhe em uma empresa de vigilância e sua principal responsabilidade como Cientista de Dados é treinar um classificador para detectar ladrões em imagens de vigilância: provavelmente está tudo bem se seu classificador tiver apenas 30% de precisão, desde que tenha 99% de recall.

Nesse caso, os seguranças receberiam alguns alertas falsos, mas quase todos os ladrões serão pegos.

Em resumo, você não pode ter as duas métricas altas ao mesmo tempo, aumentar a precisão reduz o recall e vice-versa.

Portanto, como escolher o valor para as métricas de Precisão e para o Recall?

▼ 3. Curva Precision x Recall

Os classificadores de Machine Learning, originalmente, classificam as observações do conjunto de dados como a probabilidade de pertencer a uma classe ou a outra.

É de responsabilidade do Cientista de Dados definir um limiar ou “threshold” que atribua a observação para uma classe ou outra, dependendo do valor definido. Por exemplo, se o valor do limiar é de 0.80, todas as observações com a probabilidade acima de 80% pertencem a classe A, enquanto as observações abaixo de 80% pertencem a classe B.

Os valores das métricas de precision e recall mudam de acordo com o valor do limiar de classificação definido pelo Cientista de Dados. Portanto, precisamos de uma

ferramenta visual para ajudar na escolha desse threshold.

Uma das ferramentas é a curva de Precision versus Recall e a curva de Threshold versus Precision/Recall, como veremos a seguir.

▼ 3.1 Curva Precision x Recall no Python

```
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import f1_score
from sklearn.metrics import auc
from matplotlib import pyplot as plt

# generate 2 class dataset
X, y = make_classification(n_samples=1000, n_classes=2, random_state=1)

# split into train/test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=2)

# fit a model
model = LogisticRegression(solver='lbfgs')
model.fit(X_train, y_train)

# predict probabilities
yhat_test = model.predict_proba(X_test)
yhat_test = yhat_test[:, 1]
yhat = model.predict(X_test)

# predict class values
precision, recall, thresholds = precision_recall_curve(y_test, yhat_test)
f1, auc = f1_score(y_test, yhat), auc(recall, precision)

# summarize scores
print('Model: f1=%.3f auc=%.3f' % (f1, auc))

# plot the precision-recall curves
plt.plot(recall, precision, marker='.', label='Model')
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.legend()
plt.show()
```

▼ 3.2 Curva Threshold, Precision x Recall no Python

```
plt.plot( thresholds, precisions[:-1], 'b--', label='Precision' );  
plt.plot( thresholds, recalls[:-1], 'g-', label='Recall' );  
plt.xlabel( 'Thresholds' );  
plt.ylabel( 'Precison, Recall' );  
  
plt.legend()  
plt.grid()
```

▼ 4. Resumo

1. A métricas de Precision e Recall

▼ 5. Próxima aula