

ANA RAQUEL

POSTECH

DATA ANALYTICS
MACHINE LEARNING AVANÇADO

AULA 06

SUMÁRIO

O QUE VEM POR AÍ?	3
CONHEÇA SOBRE O ASSUNTO	4
HANDS ON	10
O QUE VOCÊ VIU NESTA AULA?	11
REFERÊNCIAS.....	12

O QUE VEM POR AÍ?

Você aprendeu sobre vários tipos de algoritmos e o poder de cada um deles, mas como podemos avaliá-los de forma precisa? Será que existem as métricas certas para cada tipo de problema ou apenas a acurácia já me traz algum resultado? Nesta aula você vai conhecer as principais métricas de validação de algoritmos e suas utilidades.

CONHEÇA SOBRE O ASSUNTO

Avaliar um classificador é, muitas vezes, significativamente mais complicado do que um modelo regressor. Vamos entender o motivo conhecendo as principais métricas de validação: **Accuracy, Precision, Recall e F1 Score**.

Todas essas medidas são obtidas com base na **matriz de confusão**. A matriz de confusão, basicamente compara os acertos e erros entre os valores reais versus os valores preditos. Cada linha da matriz de confusão apresenta um valor real, enquanto cada coluna uma classe prevista.

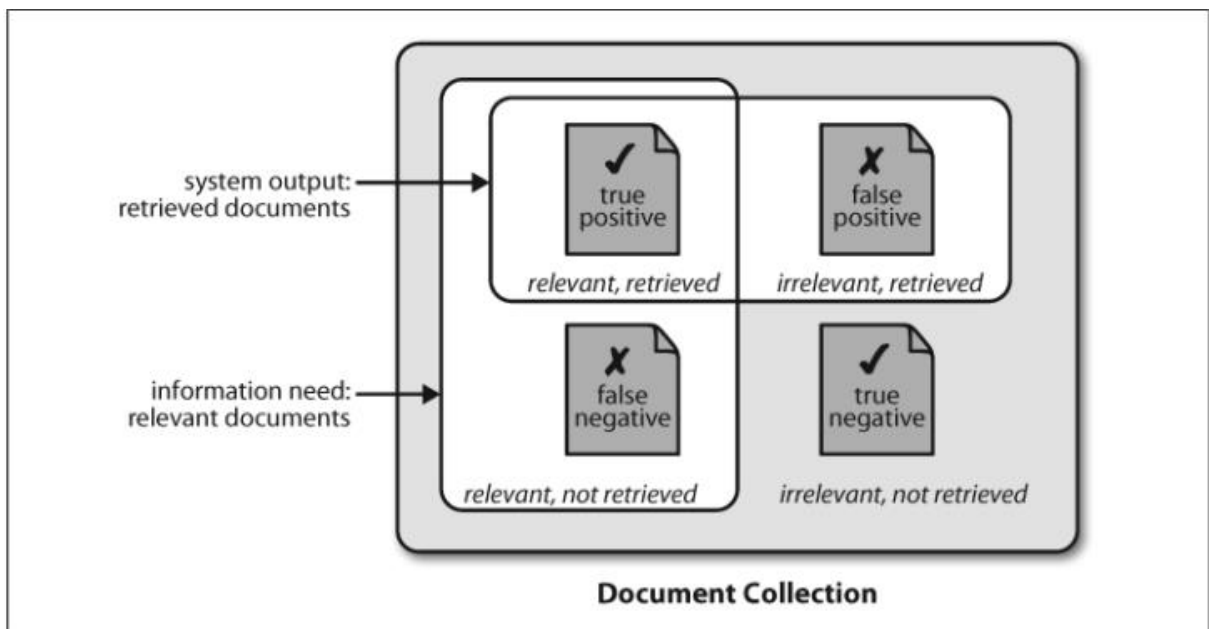


Figura 1 - Exemplo de matriz de confusão.

Fonte: Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit (2009).

Vamos identificar com um exemplo prático no Python!

Com a matriz a seguir, estamos procurando validar um modelo classificador KNN, que obteve o seguinte resultado ao prever três tipos de classes de doenças ortopédicas diferentes:

```

from sklearn.metrics import confusion_matrix
from sklearn.metrics import ConfusionMatrixDisplay

matriz_confusao = confusion_matrix(y_true = y_test,
                                   y_pred = y_predito,
                                   labels=['Disk Hernia', 'Normal', 'Spondylolisthesis'])

# plotando uma figura com a matriz de confusao
figure = plt.figure(figsize=(15, 5))

disp = ConfusionMatrixDisplay(confusion_matrix = matriz_confusao,
                              display_labels=['Disk Hernia', 'Normal', 'Spondylolisthesis'])
disp.plot(values_format='d')

```

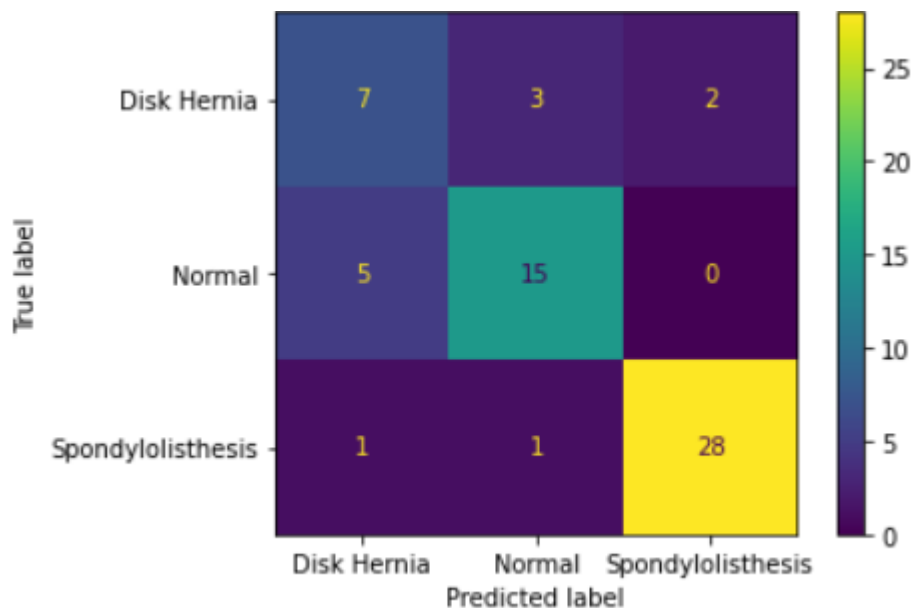


Figura 2 - Exemplo de matriz de confusão
 Fonte: Elaborado pela Autora (2023)

Perceba que, na matriz, temos as linhas sendo representadas por “**True Label**”, enquanto as colunas são as classes previstas em “**Predicted Label**”. Podemos realizar a validação dos acertos comparando cada linha com sua respectiva

coluna. Vamos começar com a classe “Disk Hernia”: se você analisar a coluna “Disk Hernia” com a linha “Disk Hernia”, irá observar que o modelo nos disse que em sete casos que são casos reais de hérnia de disco, o modelo previu de forma correta, e se compararmos a linha “Disk Hernia” com “Normal” e “Spondylolisthesis”, é possível observar que temos alguns falsos positivos:

- Em três casos que realmente são do tipo “Disk Hernia”, o modelo previu que é da classe “Normal”.
- Em dois casos que realmente são do tipo “Disk Hernia”, o modelo previu que é da classe “Spondylolisthesis”.

Dentro da matriz, é possível encontrar algumas afirmações do algoritmo, tais como:

- **Verdadeiro Positivo:** que é da classe e foi corretamente classificado;
- **Verdadeiro Negativo:** que não é da classe e não foi classificado dentro da classe;
- **Falso Positivo:** não pertence à classe, mas foi classificado dentro da classe.
- **Falso Negativo:** que é da classe, mas foi classificado fora dela.

Esse processo de validação de erros e acertos pode ser identificado como a métrica Accuracy (Acurácia). A Accuracy é a métrica de validação de algoritmos mais clássica que existe! A Acurácia mede a porcentagem de entradas no conjunto de teste que o classificador rotulou corretamente. Por exemplo, se eu estou testando 100 instâncias e meu modelo acertou 80 delas, minha acurácia é de 80%.

Podemos utilizar a ferramenta **classification_report** do Scikit-Learn para validar um conjunto de métricas importante para validar nosso modelo:

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_predito))
```

```
print(classification_report(y_test, y_predito))
```

	precision	recall	f1-score	support
Disk Hernia	0.54	0.58	0.56	12
Normal	0.79	0.75	0.77	20
Spondylolisthesis	0.93	0.93	0.93	30
accuracy			0.81	62
macro avg	0.75	0.76	0.75	62
weighted avg	0.81	0.81	0.81	62

Figura 3 - Exemplo de classification report.

Fonte: Elaborado pela Autora (2023)

Observe que o **classification_report** traz de fato um relatório de métricas, contendo cada uma das principais métricas e sua atribuição por classe. Dentro do `classification_report`, também podemos encontrar as seguintes métricas:

- **Precisão**, que indica quantos dos itens que identificamos foram relevantes, é $TP/(TP+FP)$.
- **Recall**, que indica quantos dos itens relevantes que identificamos, é $TP/(TP+FN)$.
- **F-Score**, que combina a precisão e a recuperação.
- **Pontuação única**, é definida como a média harmônica da precisão e revocação.

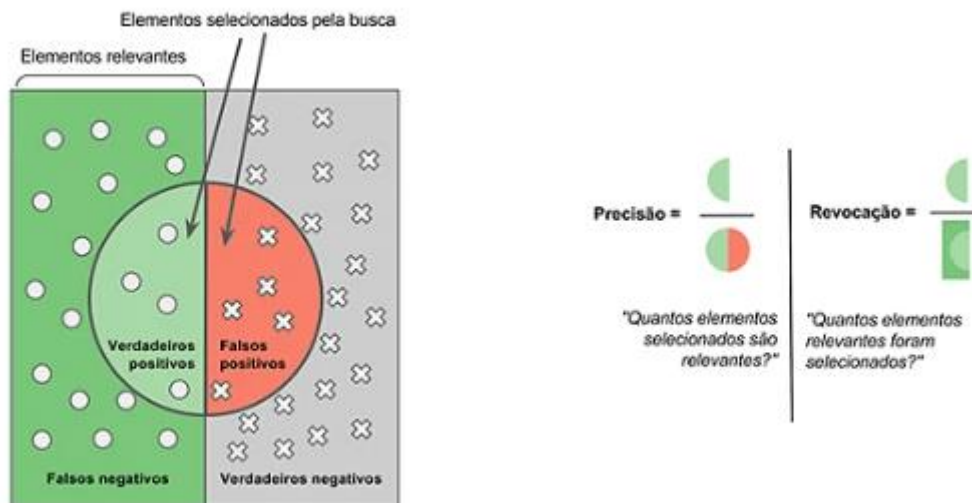


Figura 4 - Exemplo de precisão e revocação.
Fonte: Elaborado pela Autora (2023)

Ao interpretar a pontuação de precisão de um classificador, é importante considerar a frequência dos rótulos de classes individuais no conjunto de testes. Se observarmos no resultado da Figura 5 – “Classificação”, podemos identificar que o modelo classificou muito bem as classes do tipo “Spondylolisthesis”, mas para as demais classes o algoritmo não foi tão preciso assim. Você percebeu por qual motivo é importante validar o escopo da classe?

```
print(classification_report(y_test, y_predito))
```

	precision	recall	f1-score	support
Disk Hernia	0.54	0.58	0.56	12
Normal	0.79	0.75	0.77	20
Spondylolisthesis	0.93	0.93	0.93	30
accuracy			0.81	62
macro avg	0.75	0.76	0.75	62
weighted avg	0.81	0.81	0.81	62

Figura 5 – Classificação
Fonte: Elaborado pela autora (2023)

Caso você busque uma métrica harmônica entre a revocação e a precisão, escolha utilizar a métrica **F1 Score**.

HANDS ON

Agora, chegou o momento de ver, na prática, como podemos utilizar as métricas de validação de algoritmos a favor da identificação da performance do modelo.

Para essa aula, temos um notebook para você. Acesse abaixo:

- [Notebook 1](#)

Além disso, também disponibilizamos a bases de dados, para te ajudar com os estudos e exercícios.

- [Base de dados 1](#)

O QUE VOCÊ VIU NESTA AULA?

Métrica de validação; acurácia; precisão; revocação; F1 score.

Daqui em diante, é importante que você replique os conhecimentos adquiridos para fortalecer mais suas bases e conhecimentos.

IMPORTANTE: não esqueça de praticar com o desafio da disciplina, para que assim você possa aprimorar os seus conhecimentos!

Você não está sozinho(a) nesta jornada! Te esperamos no Discord e nas *lives* com os nossos especialistas, onde você poderá tirar dúvidas, compartilhar conhecimentos e estabelecer conexões!

REFERÊNCIAS

DOCUMENTAÇÃO SCIKIT-LEARN. Disponível em: <<https://scikit-learn.org/stable/>>. Acesso em: 11 abr 2023.

GÉRON, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition. [s.l.]: O'Reilly Media, Inc., 2019.

PALAVRAS-CHAVE

Palavras-Chave: Métricas De Validação; Acurácia; Precisão; Revocação; F1 Score.

The background is a dark blue field filled with numerous small, light blue dots. Overlaid on this are several large, wavy, translucent lines in shades of blue, yellow, and red. These lines flow from the left side towards the right, creating a sense of motion. Scattered throughout the composition are various geometric shapes: a circle with the number '7' inside, a small circle, an 'X' mark, a small circle, and a hexagon.

POSTECH