Técnicas de Machine Learning

UNIDADE 05

Entendendo os resultados: como saber se a máquina aprendeu ou fingiu que aprendeu

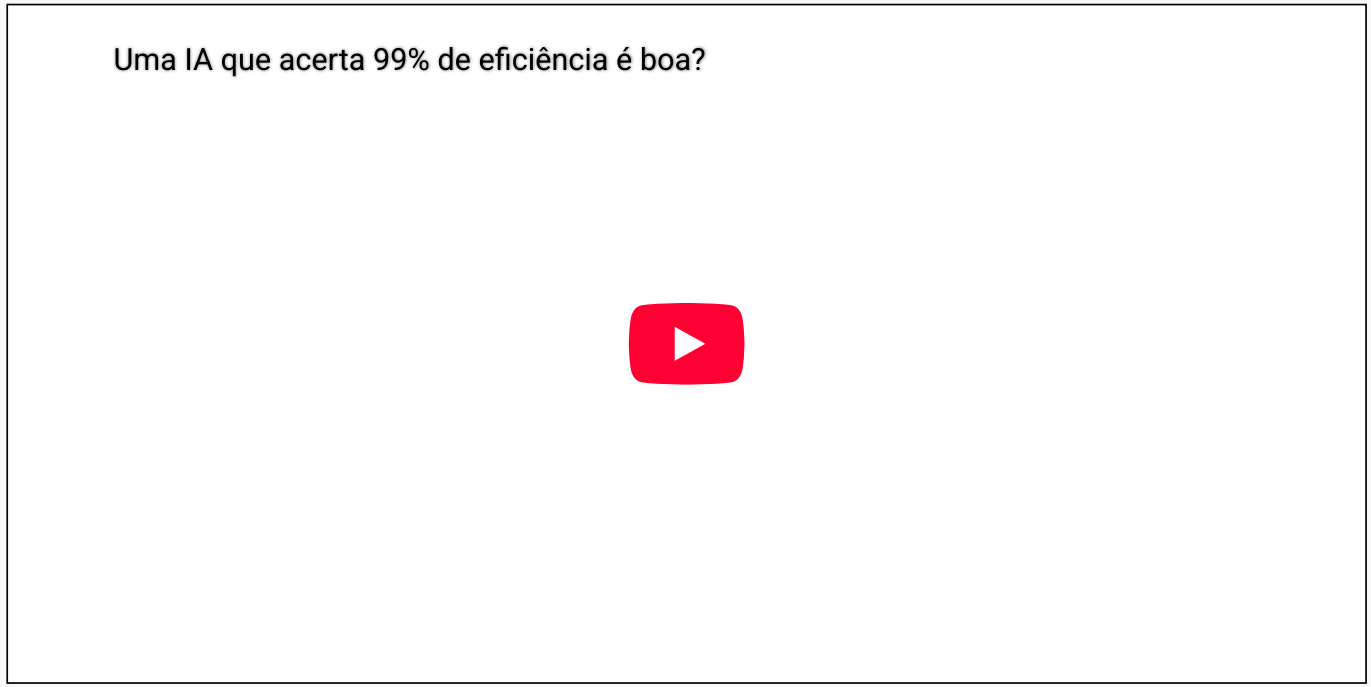
E AÍ, DEU CERTO?

Até o momento, tivemos a intenção de demonstrar como um algoritmo de ML é treinado. Você pode ter percebido, além disso, que buscamos repetir alguns passos para que esse processo ficasse claro na sua cabeça. Isso inclui, por exemplo:

- Tratamento da sua base de dados.
- Escolha de uma determinada técnica (e, para isso, usamos bibliotecas como o Scikit-learn, LightGBM, XGBoost e outros).
- Para o treinamento, separamos a nossa base utilizando o *train_test_split.*
- Vimos as previsões com uma base separada de testes.

Você pode ter notado, também, que sempre paramos o nosso processo com a geração das predições e isso sempre acontecia com uma função chamada **predict()**. Por outro lado, uma pergunta que geralmente é feita por gestores e por pessoas que não são da área é a seguinte:

> Opa, beleza, aí? Deu tudo certo com a Inteligência Artificial que você está inventando? Qual é a **acurácia** do modelo?



A “acurácia” ou a “assertividade” do modelo são termos difíceis. Existem várias métricas para entender se o modelo é **bom** ou não. Na verdade, isso é parte da avaliação do modelo e será o foco da nossa discussão. Existem, além disso, métricas para modelos de regressão e métricas para modelos de classificação. Veremos ambas aqui.

Figura 1 – Processo de ML

Fonte: adaptado de LENZ (2020).

ENTENDENDO AS MÉTRICAS

O fato de falarmos em **métricas** pressupõe em números. Para ficar mais fácil, vamos pensar em comida. Mais especificamente, em ***cookies***. Isso porque eu gosto muito de *cookies* e não consigo pensar em um outro alimento agora. E, por um grande acaso da sorte, após saber que eu gosto desse alimento, você decide me mandar uma mensagem dizendo que você é a pessoa que faz os **melhores** *cookies* da cidade.

Eu, querendo tirar vantagem da sua afirmação, digo que duvido de você e que só acreditarei quando assar alguns para mim. Na sequência, você, confiante dos seus dotes culinários, prepara cinco *cookies*.

Agora, temos alguns problemas em relação às nossas expectativas:

1. Na minha cabeça, um bom *cookie* possui gotas de chocolate, tem cerca de 10cm de diâmetro, 1cm de espessura e não é crocante. As gotas de chocolate devem ser cremosas e misturadas com a massa.
2. Na sua cabeça, um bom *cookie* possui o tamanho de uma bolacha recheada (ou biscoito?), é fino e crocante. As gotas de chocolate, por sua vez, ficam em cima e não no meio da massa do *cookie*.

Figura 2 – Quem diria não para essas delícias?

#ParaTodosVerem

Fonte: ©Lindsey Savage/Unsplash.

Percebe que as descrições são diferentes e são perfeitamente válidas? Por outro lado, é provável que eu ficaria decepcionado com seus *cookies,* porque eles estariam bem diferentes do que eu estava imaginando. Além disso, você também não se sentiria muito feliz, porque eu não teria a reação que você esperava.

Nesse caso, eu ou você teríamos culpa de algo? Diria que nenhum de nós seríamos culpados. Dessa forma, observe que tanto a minha opinião quanto a sua são apenas opiniões, ou seja, coisas subjetivas nas quais cada um pode ter um posicionamento e cada um pode se posicionar de uma forma diferente.

Quando entramos nessa área mais subjetiva, “bom” e “ruim” são termos nada sólidos. Um mesmo modelo de IA pode ser bom para mim e ruim para você. Além disso, essas afirmações podem ser difíceis de serem defendidas sem um embasamento quantitativo – ou seja, em **números**. Lembre-se, mais uma vez, de que, até este momento, basicamente só gerávamos as previsões para a base de testes. Agora, como garantir que essas previsões são suficientemente boas?

Para isso, existem as **métricas**. Elas são valores embasados na estatística, no cálculo e nas demais disciplinas das Ciências Exatas, que fornecem valores que estabelecem se um modelo de ML está ou não está de acordo com o esperado. Quando utilizamos as métricas como base (e não somente as nossas opiniões), podemos afirmar se um modelo seria **bom** ou **ruim** para uma determinada aplicação.

Pelo seu embasamento matemático, não teremos, como objetivo, a explicação de todas as equações que governam cada uma das métricas. Na verdade, mostraremos, na prática, como elas funcionam e para que servem.

Nesse sentido, comentaremos, primeiramente, sobre as métricas **mais comuns** utilizadas para algoritmos de regressão e, em seguida, para os algoritmos de classificação. Os problemas de séries temporais, que buscam prever valores, **podem** utilizar as métricas de regressão. Por sua vez, os problemas de séries temporais, que buscam prever eventos (como **sim**/**não**), podem empregar as métricas de classificação. Vamos lá?

MÉTRICAS DE REGRESSÃO

A tabela a seguir inclui as métricas de regressão sobre as quais abordaremos nesta disciplina. Estamos disponibilizando os nomes em português, inglês e a sigla. Assim, provavelmente, você irá se deparar com muitos materiais com as descrições em inglês ou, ainda, somente com base nas siglas. Vamos nos referir, a partir de agora, somente às siglas com a intenção de acostumá-lo. Observe que incluímos, além disso, os *links* para as suas respectivas implementações no Scikit-learn (sim, o Scikit-learn também inclui métricas de avaliação dos modelos).

Métrica (português)	**Métrica (inglês)**	**Sigla**
R-quadrado	R-squared	R^2 (ou R2)
Erro médio absoluto	Mean absolute error	MAE
Erro médio quadrado	Mean squared error	MSE
Erro logarítmico médio quadrado	Mean squared logarithmic error	MSLE
Erro médio absoluto percentual	Mean absolute percentage error	MAPE

R2

1. O que é

Métrica estatística que representa quão bem seus atributos explicam a sua variável dependente (a qual chamamos de “classe” em problemas de regressão. Nos notebooks vistos até o momento, o **y**: a coluna representada pelos **y_train** e **y_test**).

2. Quando usar

Em resumo, é usado em conjunto com outras métricas. Ele pode ser útil para analisar associações entre diferentes atributos, mas não de forma sozinha. Existem boas discussões sobre o seu uso (ou, mais especificamente, seu não uso – como essa discussão armazenada na Biblioteca da Universidade da Virgínia, EUA).

3. Quando não usar

- Para comparar modelos entre diferentes *datasets*: ele só é útil para comparar modelos treinados para um mesmo *dataset* (**X_train**).
- Para analisar *datasets* com muitos atributos: toda vez que um novo atributo é adicionado, o R2 acaba aumentando um pouco. Para tentar resolver isso, existe uma técnica chamada **R2 ajustado** (*adjusted R2*), a qual não está no Scikit-learn.

4. Como interpretar

- Ele, tecnicamente, gera um valor entre 0.0 e 1.0.
- Quanto mais próximo de 1.0, melhor.
- Em problemas do mundo real, é muito difícil termos um R2 alto (ou seja, 0.9 ou mais): tome cuidado se isso ocorrer para acabar não sendo enganado por um valor suspeitosamente alto.

- Um R2 baixo não significa, também, que o seu modelo é, necessariamente, ruim: existem algumas áreas como comercial (vendas) e logística, em que existem fatores externos aos quais são desconhecidos pelo seu modelo

---

A partir de agora, falaremos de **erro**. Um **erro** é a diferença entre um valor real e a previsão. Por exemplo: vamos supor que criamos um algoritmo para prever a temperatura em nossa cidade para os próximos dias. Se previmos **20** graus e o real foi **25** graus, o erro foi **5.** Se previmos **15** graus e o real foi **13** graus, o erro foi de -**2**.

MAE

1.O que é

Em resumo, é a média de todos os erros, desconsiderando seu sinal (não importando se é um erro para mais ou para menos: um erro é um erro).

2. Quando usar

Quando há uma relação proporcional entre o valor do erro e o seu **impacto**: se o erro dobrar, haverá, também, o dobro do **impacto**? Se sim, o MAE pode ser uma boa opção. No mundo financeiro, isso é comum (ex.: perder 10% é duas vezes pior do que perder 5%).

3. Quando não usar

- Em casos em que você quer penalizar os erros dos outliers (ou seja, valores muito abaixo ou muito acima da média).
- Quando a relação entre erro e impacto não são proporcionais: se errar a predição da sua nota no final da disciplina – de 6 para 7 pontos –, isso pode ser a diferença entre sua aprovação ou reprovação

4. Como interpretar

Quanto mais próximo de 0, melhor.

MSE

1. O que é

A média de todos os erros elevados ao quadrado.

2. Quando usar

Quando você quer garantir que erros grandes sejam capturados: suponhamos que ele teve pequenos erros em 100 predições, mas errou **muito** a 101ª previsão. Se você quer que essa 101ª previsão jogue o seu erro lá para cima (evidenciando que o seu modelo teve um erro grotesco, ainda que tivesse um desempenho relativamente bom em outras 100 oportunidades), o uso do MSE pode ser uma opção

3. Quando não usar

Quando você **não** deseja penalizar erros grandes

4. Como interpretar

Quanto mais próximo de 0, melhor

MSLE

1.O que é

- É similar ao MSE, mas utiliza logaritmos para determinar o erro.
- Ele pune mais os erros que são para baixo do que erros para cima (ex.: se a temperatura, hoje, foi de 20 °C, uma previsão de 18 °C terá um MSLE **maior** do que se tivesse previsto 22 °C).
- Busca determinar o erro **proporcional** ao valor (um erro de 1 real para quem tem 1000 no bolso pesa bem diferente do que um erro de 1 real para quem tem 10 reais no bolso, não é?).

2. Quando usar

Quando você se importa em penalizar a **proporção** do erro.

3. Quando não usar

- Quando você não está trabalhando com dados contínuos.
- Quando você não deseja pesar, de formas diferentes, erros para cima e erros para baixo.

4. Como interpretar

Quanto mais próximo de 0, melhor.

MAPE

1. O que é

É a média das porcentagens dos erros (ex.: se o valor real era 100 e a previsão foi 90, o erro foi 0.10. Se o valor real era 100 e a previsão foi 110, o erro também foi 0.10).

2. Quando usar

Quando você deseja que alguns erros percentuais potencialmente destoantes dos demais (ou seja, muito altos) afetem a métrica.

3. Quando não usar

- Quando você possui muitos *outliers*.
- Quando você possui muitos zeros.

4. Como interpretar

Quanto mais próximo de 0, melhor.

(Algumas) variações

1. MSE

a. **RMSE (Root Mean Squared Error):** o MSE retorna um valor que não é a métrica do que você está medindo (ou seja, em graus Celsius, dólares, reais, centímetros ou qualquer outra unidade de medida do seu problema) – o RMSE, na verdade, é a raiz quadrada do MSE e deixa os valores em um formato mais compreensível para humanos.
b. **NRMSE (Normalized Root Mean Squared Error):** o erro é normalizado, convertendo-o essencialmente para uma porcentagem (cuidado: a lógica por trás não faz dele a mesma coisa que o MAPE!).

2. MSLE

a. **RMSLE (Root Mean Squared Logarithmic Error):** é tão somente a raiz quadrada do MSLE.

Exemplo

Imaginemos um algoritmo que prevê o saldo em conta-corrente de um conjunto de pessoas. Por questões de visualização, mostraremos somente duas colunas: a primeira, SaldoConta, possui o valor real. A segunda, Predicao, contém as predições. Observe, nas colunas ao lado, qual seria o valor dos erros dependendo de cada métrica. Além disso, estamos usando todas as métricas que estão disponíveis no Scikit-learn.

SaldoConta	Predicao	MAE	MSE	RMSE	MSLE	MAPE
150150	15	150135.000	22540518225.000	150135.000	83.664	1.000
100	10	90.000	8100.000	90.000	4.916	0.900
10	100	90.000	8100.000	90.000	4.916	9.000
1	1	0.000	0.000	0.000	0.000	0.000
0	0	0.000	0.000	0.000	0.000	0.000
0	10	10.000	100.000	10.000	5.750	45035996273704960.000
0	0	0.000	0.000	0.000	0.000	0.000
50	49	1.000	1.000	1.000	0.000	0.020
500	490	10.000	100.000	10.000	0.000	0.020
500	510	10.000	100.000	10.000	0.000	0.020
5000	5001	1.000	1.000	1.000	0.000	0.000
50000	50001	1.000	1.000	1.000	0.000	0.000
500000	500500	500.000	250000.000	500.000	0.000	0.001

Ao validarmos um **modelo**, não observamos os erros de cada uma das instâncias, mas sim do total delas. Baseando-se em todas as instâncias anteriores, os erros seriam os seguintes:

1. R2 para todas as instâncias: 0.905.
2. MAE para todas as instâncias: 11603.692.
3. MSE para todas as instâncias: 1733906517.538.

4. RMSE para todas as instâncias: 41640.203.
5. MSLE para todas as instâncias: 7.634.
6. MAPE para todas as instâncias: 3464307405669613.000.

O que vale a pena observarmos

1. Tínhamos, logo na primeira linha, um erro: um dos valores a serem previstos era R$ 150150. No entanto, prevemos somente R$ 15 como resultado. Isso resultou em um MSE astronômico (e os demais valores também ficaram altos). Observe, ainda, que o MAPE ficou travado em "1.000".
2. Por outro lado, quando o valor real era 0 e a previsão foi 10 (somente R$ 10 de diferença), tivemos um MAPE gigantesco – ainda maior do que o erro dos R$ 150150.
3. Perceba o efeito que esses dois casos tiveram no MSE e o MAPE, considerando todas as instâncias.

Observe que o MAPE ficou gigantesco, principalmente por conta daquele saldo zerado, no qual houve uma predição de 10. Agora, o que aconteceria se a predição só para aquela linha fosse um zero? Dessa forma, os resultados seriam:

SaldoConta	Predicao	MAE	MSE	RMSE	MSLE	MAPE
150150	15	150135.000	22540518225.000	150135.000	83.664	1.000
100	10	90.000	8100.000	90.000	4.916	0.900
10	100	90.000	8100.000	90.000	4.916	9.000
1	1	0.000	0.000	0.000	0.000	0.000
0	0	0.000	0.000	0.000	0.000	0.000
0	0	0.000	0.000	0.000	0.000	0.000
0	0	0.000	0.000	0.000	0.000	0.000
50	49	1.000	1.000	1.000	0.000	0.020
500	490	10.000	100.000	10.000	0.000	0.020
500	510	10.000	100.000	10.000	0.000	0.020
5000	5001	1.000	1.000	1.000	0.000	0.000

50000	50001	1.000	1.000	1.000	0.000	0.000
500000	500500	500.000	250000.000	500.000	0.000	0.001

1. R2 para todas as instâncias: 0.905.
2. MAE para todas as instâncias: 11602.923.
3. MSE para todas as instâncias: 1733906509.846.
4. RMSE para todas as instâncias: 41640.203.
5. MSLE para todas as instâncias: 7.192.
6. MAPE para todas as instâncias: 0.843.

O que vale a pena observarmos

1. No geral, o MSE continua altíssimo, mas o MAPE abaixou muito apenas com aquele erro.
2. Perceba que o R2, o RMSE e o MSLE não mudaram muito: qual é o seu entendimento ao comparar esses resultados com as descrições dos erros mencionados anteriormente?

E se a previsão da primeira linha fosse 155150 ao invés de 15? Observe a seguir.

SaldoConta	Predicao	MAE	MSE	RMSE	MSLE	MAPE
150150	155150	1050.000	1102500.000	1050.000	0.000	0.007
100	10	90.000	8100.000	90.000	4.916	0.900
10	100	90.000	8100.000	90.000	4.916	9.000
1	1	0.000	0.000	0.000	0.000	0.000
0	0	0.000	0.000	0.000	0.000	0.000
0	0	0.000	0.000	0.000	0.000	0.000
0	0	0.000	0.000	0.000	0.000	0.000
50	49	1.000	1.000	1.000	0.000	0.020

500	490	10.000	100.000	10.000	0.000	0.020
500	510	10.000	100.000	10.000	0.000	0.020
5000	5001	1.000	1.000	1.000	0.000	0.000
50000	50001	1.000	1.000	1.000	0.000	0.000
500000	500500	500.000	250000.000	500.000	0.000	0.001

1. R2 para todas as instâncias: 1.000.
2. MAE para todas as instâncias: 134.846.
3. MSE para todas as instâncias: 105300.231.
4. RMSE para todas as instâncias: 324.500.
5. MSLE para todas as instâncias: 0.756.
6. MAPE para todas as instâncias: 0.767.

O que vale a pena observarmos

1. O R2 aumentou ainda mais – de fato, parece que está perfeito. Considerando esses exemplos e comparando com os demais, você entende que o R2 funcionaria como uma boa métrica de comparação?
2. O MSE abaixou bastante só ao arrumar o *outlier*.
3. Observe que o MSLE e o MAPE abaixaram, mas em uma proporção menor.
4. Observe os valores dos erros. Qual é a interpretação que você possui? O que seria um RMSE bom? Ou um MAPE bom?
5. Podem existir métricas que podem funcionar bem para certos *datasets* e não serem tão úteis para outros. De forma geral, há uma certa predileção pelo RMSE (ou NRMSE), MAPE e suas variantes. Logo, eles podem ser bons pontos de partida. Por outro lado, reforçamos: sempre podem existir exceções e os profissionais podem, também, preferir outras métricas.

IMPORTANTE

***Atenção**: as métricas devem ser usadas ao analisar um **dataset** inteiro (no caso, a base de testes), e não instância por instância. Mostramos, aqui, das duas formas **somente** para que você observe o resultado das métricas com diferentes valores de predição.*

MÉTRICAS DE CLASSIFICAÇÃO

A tabela a seguir inclui as métricas de classificação. Contudo, utilizaremos a mesma lógica de explicação das métricas de regressão. Logo, também incluímos os *links* nas siglas para as suas respectivas implementações no Scikit-learn (o Scikit-learn também inclui métricas de avaliação de **classificação** dos modelos). Além disso, vale a pena observar que 99% das vezes as pessoas se referem pelos nomes em inglês ou, ainda, pelas suas siglas em inglês.

Métrica (português)	Métrica (inglês)	Sigla
Curva Característica de Operação do Receptor	Receiver Operating Characteristic Curve	Curva ROC
Área sob a Curva Característica de Operação do Receptor	Area Under the Receiver Operating Characteristic Curve	ROC AUC
Precisão e revocação	Precision and recall	Precision-Recall
Escore F1	F1 score	F1 score
Matriz de confusão	Confusion matrix	Confusion matrix
Acurácia	Accuracy score	Accuracy (e balanced accuracy)

Antes de nos aprofundarmos nas explicações, é válido pensarmos em um exemplo. Um dos locais que contém os mais altos níveis de segurança são os **aeroportos internacionais**. Eles possuem um conjunto de controles, o que inclui, por exemplo, verificações aleatórias de segurança, verificações de passaporte, raios-X, detectores de metais e outros.

Figura 3 – Esses locais, geralmente, possuem uma comida bem cara

#ParaTodosVerem

Fonte: ©Daniel Lim/Unsplash.

Esses processos de segurança servem para encontrar pessoas com intenções maliciosas, o que pode incluir casos de tráfico de drogas, tráfico humano, atentados terroristas e outras ameaças à segurança dos passageiros. Logo, é **muito** importante filtrarmos esses casos, uma vez que estamos falando de vidas, não é mesmo? Por outro lado, sabemos que, estatisticamente falando, são pouquíssimas as pessoas que possuem essas intenções. Provavelmente, seria 1 pessoa a cada 5 mil que passam no aeroporto. E, ainda, é bem difícil conseguir saber, logo de cara, quem teria essas intenções, uma vez que ninguém tem uma bola de cristal para prever.

Agora, suponhamos que você foi contratado como o chefe de segurança do aeroporto. Você ganha bem, mas possui uma grande responsabilidade, ou seja, se algo de errado acontecer, a culpa será totalmente sua. Logo, você **não pode** deixar que nenhuma pessoa mal-intencionada passe pelos seus filtros de segurança.

Assim, como você resolveria esse problema? Como pegará uma pessoa mal-intencionada dentre milhares de outras? Vamos pensar em alguns cenários?

Vamos partir da premissa de que existem **20 mil** pessoas passando no aeroporto hoje. Dessas, **50** são mal-intencionadas. Então, **temos 19950 que seriam inocentes e 50 seriam culpadas**. No entanto, você não sabe, com antecedência, quem é quem.

1. Você é alguém que não pode assumir riscos (C1): imagine só – é um bom emprego, mas você tem a responsabilidade sobre a vida de várias pessoas. Tudo o que for preciso para barrar alguém mal-intencionado será feito!

a. **Proposta:** você decide fazer uma verificação de segurança adicional (o que chamaremos de *"pente-fino"*) para **todos os 20 mil** passageiros. Você entende que todos eles são culpados até que se prove o contrário.
b. **Resultado:** com isso, formou-se uma fila gigantesca – existem poucos profissionais para verificar tantos passageiros. Vários passageiros perderam os voos e reclamaram sobre as suas práticas para os seus superiores. Essa ação causou, ainda, um congestionamento enorme no aeroporto. Assim, na sequência, você foi demitido.

2. Você é alguém que não acredita que algum problema aconteceria novamente (C2): você acha que o aeroporto em que trabalha não seria visado por pessoas mal-intencionadas. Logo, você acredita que todos os 20 mil passageiros são inocentes.

a. **Proposta:** você decide que **todos os 20 mil** passageiros podem passar sem maiores verificações, uma vez que acredita que todos são inocentes até que se prove o contrário.
b. **Resultado:** existiam, realmente, 50 pessoas com más-intenções e que causaram, com seus crimes, danos gigantescos ao país. O tribunal achou que você compactuou com os criminosos ao facilitar a saída deles do aeroporto. Assim, você foi demitido e, inclusive, preso.

Percebe que nenhum dos extremos (C1 e C2) são legais? Assim, pensemos, agora, em outros dois casos.

1. Você prefere pecar pelo excesso do que pela falta (C3): você entende que precisa encontrar um equilíbrio entre dois extremos, ou seja, não pode causar filas gigantescas e não pode, também, deixar todo mundo passar batido sem filtro algum.

a. **Proposta:** você decide fazer algumas verificações por meio do cruzamento de perfis dos passageiros. Dos 20 mil passageiros, você verifica, adicionalmente, 1000 passageiros (bem mais do que os 50 mal-intencionados).
b. **Resultado:** com isso, algumas filas foram criadas, mas nada que tenha incomodado o fluxo do aeroporto. Dos 1000 passageiros que filtrou, você acabou pegando 45 mal-intencionados. Nada mal!

2. Você prefere testar alguns casos de forma aleatória (C4): você entende que é necessário começar algo, mas de forma cautelosa. Assim, é preferível pegar, pelo menos, uma pessoa mal-intencionada do que nenhuma.

a. **Proposta:** você decide fazer algumas verificações aleatórias de segurança. Dos 20 mil passageiros, você escolheu, aleatoriamente, 1000 passageiros.
b. **Resultado:** como escolheu aleatoriamente, dos 1000 passageiros selecionados, somente 5 deles eram mal-intencionados. Você entende que já é alguma coisa, mas ainda há um espaço de melhora?

Agora, qual dos casos (C1, C2, C3 e C4) seria o melhor? Vamos, antes de mais nada, estabelecer alguns termos. Essa ação irá nos auxiliar para entendermos melhor as métricas. Além disso, para que os conceitos fiquem mais claros, pensaremos sempre em problemas com duas classes.

a) Positivo e negativo (*positive* and *negative*, ou P e N)

São as duas classes que você deseja prever e isso depende de *dataset* para *dataset*. Pode ser verdadeiro/falso, entra/sai, aprovado/reprovado, se a imagem possui um gato/cachorro etc. No caso do aeroporto, é mal-intencionado/não é mal-intencionado. Então, a partir de agora, convencionaremos que uma pessoa mal-intencionada é um caso positivo (P) e uma pessoa, que é inocente, um caso negativo (N).

b) TP (*true positive*) e TN (*true negative*)

a. Quando um algoritmo **acertou** que uma pessoa era realmente um caso positivo, entendemos que esse é um TP.
b. Quando um algoritmo **acertou** que uma pessoa era realmente um caso negativo, entendemos que esse é um TN.

    a. Sempre queremos o máximo possível de TP e TN.

c) FP (*false positive*) e FN (*false negative*)

a. Quando um algoritmo acusou que uma pessoa era um caso positivo e, na verdade, era um caso negativo, temos um FP. É um alarme falso.
b. Quando um algoritmo acusou que uma pessoa era um caso negativo e, na verdade, era um caso positivo, temos um FN. É um erro que deixamos passar.
c. Sempre queremos zerar (quando possível) os FP e FN.
d. Dependendo do problema, é melhor termos um FP do que um FN, e vice-versa: no caso do aeroporto, o que é mais perigoso? Um FP ou um FN? No caso, certamente, seria um FN, pois deixamos passar uma pessoa mal-intencionada e ela poderá causar danos bem maiores do que um eventual FP.

Olhando os casos anteriores, temos o seguinte (preste muita atenção, uma vez que essa será a base para a explicação das métricas):

1. Para todos os casos: 50 P, 19950 N.

2. C1 (em que achou que todos eram culpados):

    a. TP: 50, FP: 19950.
    b. FN: 0, TN: 0.

3. C2 (em que achou que todos eram inocentes):

    a. TP: 0, FP: 0.
    b. FN: 50, TN: 19950.

4. C3 (em que está fazendo o cruzamento de perfis):

a. TP: 45, FP: 955.
b. FN: 5, TN: 18995.

5. C4 (em que está fazendo verificações aleatórias):

a. TP: 5, FP: 995.
b. FN: 45, TN: 18955.

Agora sim, vamos às métricas!

Accuracy

A acurácia é, para todos os efeitos, uma porcentagem de todas as vezes que **acertamos**. Ou seja, quantos TP e TN tivemos em relação ao todo:

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

Aplicando a equação anterior, para os quatro casos teríamos:

1. C1: 0.0025 (2.5%).
2. C2: 0.9975 (99.75%).
3. C3: 0.9520 (95.20%).
4. C4: 0.9480 (94.80%).

A princípio, sempre queremos a **maior** acurácia: 100% de acurácia seria um mundo perfeito, não é mesmo? Afinal, sempre acertaria as predições. Olhando os resultados anteriores, é fácil predizer, pois o C2 tem a maior acurácia e, portanto, é a melhor! 99.75% de acurácia é quase perfeito, certo? Na verdade, não! Observe a descrição do C2 – é praticamente um cenário catastrófico. Esse é, na verdade, o ponto que gostaríamos de trazer. Principalmente para casos de **classes desbalanceadas** (quando existem muito mais instâncias de uma classe do que de outra), a acurácia pode nos passar uma imagem de que tudo está perfeito quando, na verdade, não está. No caso do aeroporto, 99.75% é horrível, pois não queremos deixar todo mundo passar. Olhando para isso, o que precisaríamos, na verdade, seria de uma métrica que melhor representasse esse desequilíbrio entre essas duas classes.

Para resolver isso, temos a acurácia balanceada. Ela busca dar igual importância para o P e N:

$$BalancedAccuracy = \frac{1}{2} * (\frac{TP}{TP + FN} + \frac{TN}{TN + FP})$$

Observe como ficariam os resultados agora:

1. C1: 0.5000 (50%).
2. C2: 0.5000 (50%).
3. C3: 0.9261 (92.61%).
4. C4: 0.5251 (52.51%).

Observe que os valores mudaram completamente. C1 e C2 passaram a ter as mesmas acurácias ruins (julgar tudo como P ou julgar tudo como N é igualmente errado) e o C4 é levemente melhor do que o C1 e C2 – praticamente por sorte. Por sua vez, o C3 parece ser mais robusto do que os demais. Agora, volte às definições dos quatro casos e responda: você acha que esses valores são mais honestos em relação à realidade?

Precision and recall

Precision pode ser entendido das seguintes formas:

1. Quantas vezes o algoritmo falou que era P e **realmente era** P?
2. Quando um algoritmo prevê um P ele estaria correto quantas vezes?

Recall pode ser entendido das seguintes formas:

1. Quantas vezes **realmente era** P e o algoritmo falou que era P?
2. De todos os P possíveis, o algoritmo acertou quantas vezes?

Ambos podem soar bem parecidos, mas observe a diferença nas equações, pois um usa o FP como base das suas comparações e, o outro, o FN:

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

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

Voltemos aos 4 casos para entender os resultados.

1. C1:

    a. *Precision*: 0.0025.
    b. *Recall*: 1.0000.

2. C2:

    a. *Precision*: NaN (*not a number*; erro de divisão por zero).
    b. *Recall*: 0.0000.

3. C3:

    a. *Precision*: 0.0450.
    b. *Recall*: 0.9000.

4. C4:

    a. *Precision*: 0.0050.
    b. *Recall*: 0.1000.

Vamos, agora, interpretar esses valores. O primeiro ponto é: **quanto maior, melhor**. Com isso, já eliminamos o C2, pois o *recall* é 0 e nem conseguimos calcular o *precision*, uma vez que não temos FP e nem TP. Depois, temos o C1, pois o *recall* parece ser muito bom (ele nunca deixou passar um FN), mas o *precision* é ruim (ele achou vários FPs – achar um TP seria como encontrar uma agulha no palheiro).

Então, temos o C4. Ele tem uma *precision* bem maior do que o C1 (temos, proporcionalmente, menos FPs do que o C1), mas o *recall* é pior (uma vez que deixou passar vários FNs).

Finalmente, temos o C3. Observe que, proporcionalmente, temos um *precision* razoavelmente maior do que os demais (ou seja, suas previsões do P são mais confiáveis) e um *recall* um pouco pior do que o C1 (ou seja, deixou relativamente poucos FNs passarem).

Perceba que sempre queremos tanto o *precision* quanto o *recall* os mais próximos do 1.000 possível. Assim, analisar **só** o *precision* ou **só** o *recall* não te dará o necessário para tomar uma decisão. Logo, seria preferível analisar ambos em conjunto.

F1 score

É conhecido, também, por F-measure ou F-score e busca agregar o *precision* e o *recall* em uma única métrica. Quanto mais próximo de 1, melhor. Quanto mais próximo de 0, pior. Ele é definido por:

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Para todos os casos, teríamos:

1. C1: 0.0050.
2. C2: NaN (erro com o *precision*).
3. C3: 0.0857.
4. C4: 0.0095.

Observe que esses números não são muito bons se pensarmos que precisaríamos chegar perto de 1.000. Por outro lado, considerando somente esses quatro casos, chegamos novamente à conclusão de que o C3 seria o melhor. Perceba, então, como esses números refletem as justificativas do *precision* e do *recall* analisados anteriormente.

Confusion Matrix

Na realidade, já vimos a matriz de confusão anteriormente. Lembremos, assim, da listagem dos casos:

1. C1 (em que achou que todos eram culpados):

   a. TP: 50, FP: 19950.
   b. FN: 0, TN: 0.

2. C2 (em que achou que todos eram inocentes):

   a. TP: 0, FP: 0.
   b. FN: 50, TN: 19950.

3. C3 (em que está fazendo o cruzamento de perfis):

   a. TP: 45, FP: 955.
   b. FN: 5, TN: 18995.

4. C4 (em que está fazendo verificações aleatórias):

   a. TP: 5, FP: 995.
   b. FN: 45, TN: 18955.

A matriz de confusão apresenta os TP, FP, FN e TN em um formato de tabela:

	Casos previstos (P)	**Casos previstos (N)**
Casos reais (P)	TP	FN
Casos reais (N)	FP	TN

Sempre procuraremos um modelo em que o TP e o TN são altos, enquanto o FP e o FN são baixos. Para o C1 (quanto maiores os números, mais escura é a cor de fundo):

	Casos previstos (P)	**Casos previstos (N)**
Casos reais (P)	50	0
Casos reais (N)	19950	0

Para o C2:

	Casos previstos (P)	**Casos previstos (N)**
Casos reais (P)	0	50

Casos reais (N)	0	19950

Para o C3:

	Casos previstos (P)	Casos previstos (N)
Casos reais (P)	45	5
Casos reais (N)	955	18995

Para o C4:

	Casos previstos (P)	Casos previstos (N)
Casos reais (P)	5	45
Casos reais (N)	955	18955

O caso ideal seria onde o TP e o TN estivessem em um fundo escuro e o FP e o FN estivessem em um fundo claro. Novamente, vemos que o C3 está melhor do que os demais, mas que ainda seria possível ter algo melhor.

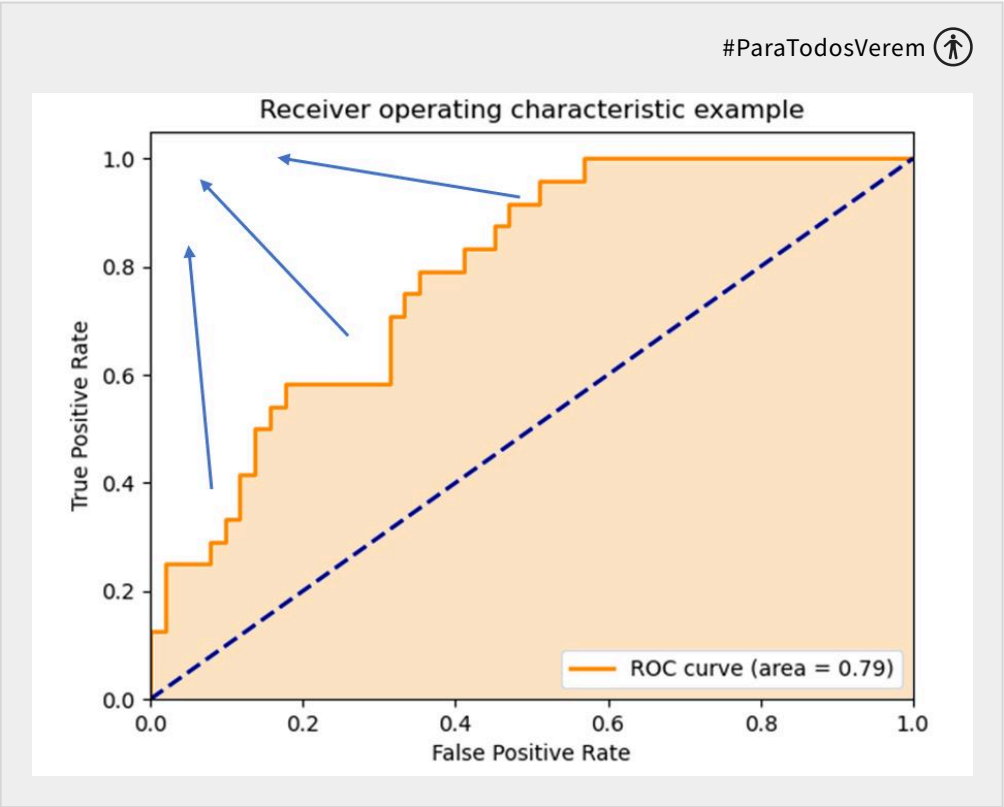
A matriz de confusão é muito útil para apresentar os resultados para usuários e analistas sem experiência com ML, uma vez que ilustra os resultados de uma forma que é mais inteligível para diversos públicos.

ROC e AUC

A curva ROC é um gráfico que busca apresentar, de uma forma visual, a *performance* de um modelo. Lembre-se de que um modelo de ML de classificação, geralmente, prevê uma *label*: sim/não; verdadeiro/falso e P/N. Por baixo dos panos, alguns algoritmos também conseguem informar a **probabilidade** da previsão (dica: procure na documentação do RandomForestRegressor do Scikit-learn por uma função chamada **predict_proba**).

Observe a imagem a seguir, adaptada da própria documentação do Scikit-learn. A linha tracejada na diagonal seria um classificador aleatório, ou seja, basicamente, um jogo de cara ou coroa. O que queremos é que a curva esteja o mais próxima possível do canto superior esquerdo (observe as três flechas apontando para essa direção). Assim, uma curva que estivesse mais próxima a esse canto teria uma *performance* melhor.

Figura 4 –Exemplo de característica de operação do receptor



Fonte: adaptado de https://scikit-learn.org/stable/modules/model_evaluation.html#roc-metrics.[/legenda]

Para avaliar essa *performance* com um número, temos o AUC, ou seja, o cálculo da área **sob** a curva. No caso, a área mais escura abaixo da curva laranja é o seu AUC. Quanto mais próximo de 1, melhor. Quanto mais próximo de 0, pior. Nesse exemplo, o AUC foi de 0.79.

MÃO NA MASSA

Observe que existe um *notebook* em python na VM, intitulado ***Semana6_Metricas***. Ele foi criado para mostrar as métricas de classificação e regressão embasadas no Scikit-learn. Para a base de classificação, utilizamos um *toy dataset* de câncer de mama. O contexto dele é bem parecido com o exemplo do aeroporto, ou seja, é muito importante evitarmos FNs e FPs. No entanto, os FNs possuem uma importância enorme (nesse caso, dizer para um paciente que ele não teria câncer quando, na realidade, teria). Assim, analise as diferentes métricas criadas com diferentes configurações do RandomForestClassifier. Busque interpretar, além disso, os resultados com os comentários feitos até o momento.

Após analisar o *notebook*, assista ao vídeo *Escolhendo o melhor modelo.* Nele, é necessário escolher um melhor modelo de ML, com base em um conjunto de modelos pré-treinados e comparando, ao mesmo tempo, diferentes métricas.

Escolhendo o melhor modelo

Até as semanas anteriores, preocupávamo-nos em somente gerar as predições, mas não em comparar a *performance* dos modelos. Vamos ver, agora, como diferentes métricas podem ser utilizadas em conjunto para escolher o melhor modelo? Mas, lembre-se: aplique as métricas sempre na base de testes para evitar de tomar uma decisão errada.

CONCLUSÃO

Nesta unidade, aprofundamo-nos no tópico de métricas de modelos. Assim, você teve contato com as métricas mais comuns de avaliação de modelos de ML – sejam eles de regressão ou de classificação. A ideia não é a de que utilizemos somente uma métrica para todos os modelos no futuro, pois existem casos nos quais o MAPE pode funcionar bem e, em outros, o MSLE pode funcionar melhor. O mesmo vale para a classificação, uma vez que a "acurácia" é um termo perigoso e pode causar uma impressão errada (o exemplo do aeroporto confirma essa afirmação).

REFERÊNCIAS BIBLIOGRÁFICAS

FACELI, K. *et al.* **Inteligência artificial**: uma abordagem de aprendizado de máquina. 2. ed. Rio de Janeiro: LTC, 2021. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788521637509/. Acesso em: 21 out. 2024.

HUYEN, C. **Projetando sistemas de Machine Learning**: processo interativo para aplicações prontas para produção. Rio de Janeiro: Editora Alta Books, 2024.

RUSSELL, S.; NORVIG, P. **Inteligência Artificial**. 4. ed. Rio de Janeiro: LTC, 2024.