Técnicas de Machine Learning

UNIDADE 08

Dominando o tempo: previsão de séries temporais

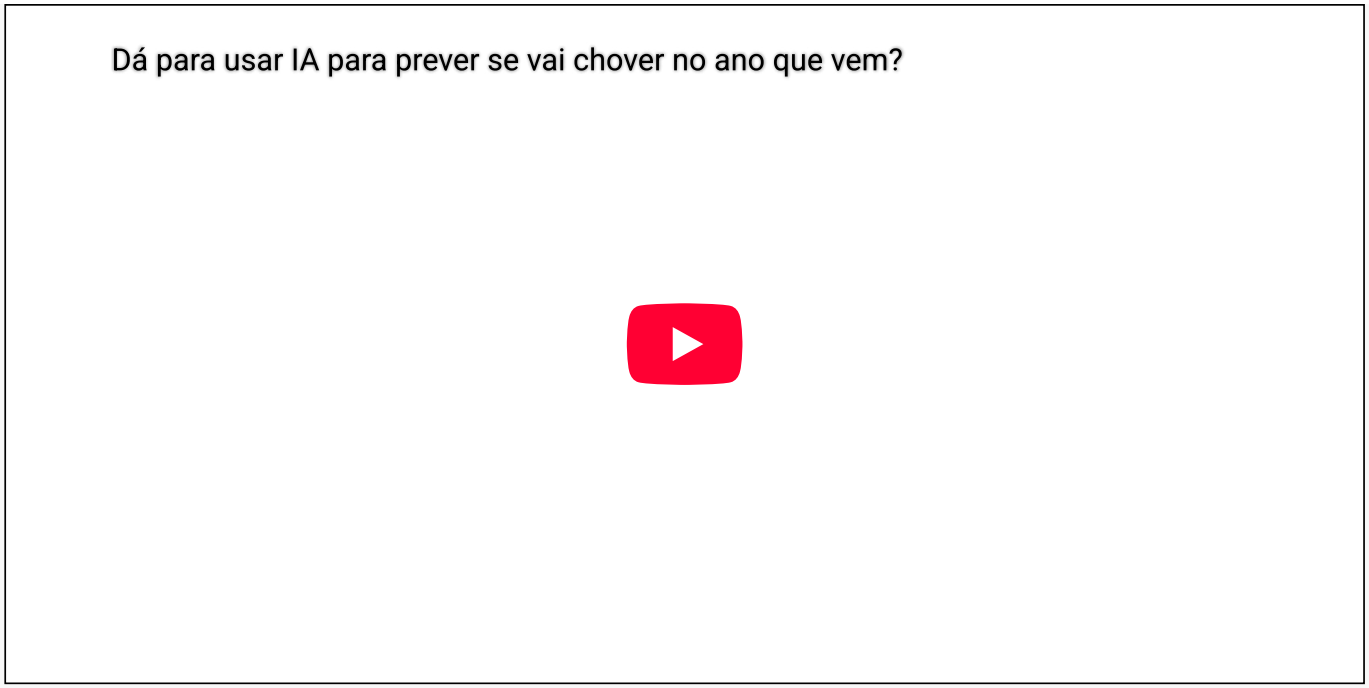
ANALISANDO A NOSSA LINHA DO TEMPO

Olha só: na penúltima unidade tivemos a oportunidade de introduzir a você sobre as cinco tribos. Também conversamos sobre a aprendizagem supervisionada e as suas principais vertentes: classificação e regressão. Conversamos sobre a forma na qual realizamos o treinamento de um algoritmo de aprendizagem supervisionada e até introduzimos dois subtipos bem interessantes de algoritmos: os comitês de algoritmos (*ensembles*) e as séries temporais.

Na última unidade revisamos três etapas do processo de ciência de dados: **preparação dos dados**, **seleção do modelo** e **treinamento do modelo**. Disponibilizamos alguns códigos e exemplos de criação de algoritmos utilizando bibliotecas como o scikit-learn, XGBoost, LightGBM e Prophet. Além disso, trabalhamos em uma atividade com a intenção de criar um algoritmo de classificação ou de regressão.

Figura 1 – Processo de ML

Fonte: adaptado de Lenz (2020).



Dito isso, agora vamos nos aprofundar exclusivamente nas **séries temporais** sobre as quais já começamos a conversar anteriormente. Em uma sociedade na qual se torna cada vez mais relevante prever o que acontecerá no futuro, faz-se importante conhecermos como esses algoritmos funcionam. Geralmente, pessoas com um conhecimento maior em disciplinas como estatística e econometria demonstram um grande conhecimento nessa área – de fato, a quantidade de técnicas de séries temporais é tão grande que não é raro vermos pessoas estudando e se aprofundando durante vários anos em níveis de mestrado e doutorado somente nessa temática (quer dizer – isso também se aplica às outras técnicas que também já vimos aqui). Logo, o recado é que temos como intenção apresentar a você o tema e algumas técnicas de previsão de séries temporais. Por outro lado, é igualmente importante que você saiba que existem várias outras técnicas que também possam lhe auxiliar nesse sentido – técnicas essas que podem também utilizar conhecimentos de engenharia, matemática e áreas afins em disciplinas como cálculo, por exemplo. Portanto, não se assuste: tentaremos deixar o conteúdo mais amigável para você. Vamos lá?

DANDO UMA PAUSA NO TEMPO

Não sei se percebeu, mas quando trabalhamos com algoritmos de aprendizagem supervisionada ou não supervisionada, as **instâncias** são independentes entre si. Note que não importava muito a ordem das instâncias. Voltemos aos *datasets* ***iris*** ou ***wine***: percebeu que nunca nos importamos quanto à data de coleta das flores, ou se os valores foram coletados de manhã ou de tarde? Uma flor é totalmente independente de outra flor; um vinho é diferente do outro; e uma pessoa é totalmente diferente da outra.

Se pensarmos no mundo real, o tempo em si pode, sim, ter uma participação: um vinho de 30 anos pode ter propriedades diferentes de um vinho de 3 meses. Um botão de flor também poderá ter tamanhos diferentes de uma flor próxima ao final da sua vida, não concorda? Até mesmo um *dataset* que mede as condições financeiras das pessoas poderá ter resultados bem diferentes se medirmos isso na época do PLR (participação nos lucros ou resultados) ou do décimo-terceiro salário, ou se medíssemos na época de compra de material escolar, pagamento do IPVA ou IPTU, e assim por diante. Ora: até mesmo a condição financeira das pessoas muda **dentro de um único mês** (o dia do pagamento é uma coisa e no final do mês é outra, não é?).

Então, o tempo pode ser capaz de influenciar os *datasets* de uma forma geral. Se trabalhamos com quaisquer problemas de aprendizagem supervisionada ou de aprendizagem não supervisionada, é interessante termos uma **padronização** no tempo. Vamos imaginar os seguintes casos:

1. Se estivéssemos criando um algoritmo para **aprovar ou reprovar a emissão de um cartão de crédito** para novos clientes, precisaríamos ter uma base cadastral colhida em uma mesma época (imagine se coletássemos todo o histórico de 1994 até hoje: uma pessoa ganhando R$ 2 mil em 2000 estaria em uma condição social diferente de uma outra pessoa ganhando R$ 2 mil nos dias de hoje, não concorda?).
2. Se estivéssemos tentando **prever o preço de venda de uma casa** a partir de suas características como bairro, idade da casa, número de vagas na garagem, banheiros e quartos precisaríamos de uma base com todos os preços ajustados pela inflação, não acha? Caso contrário, poderíamos cair no perigo de prever um preço bem desatualizado de uma casa.
3. Se estivéssemos trabalhando com um *dataset* de um histórico de pacientes de um hospital para prever se novos pacientes **teriam ou não uma determinada doença** precisaríamos da idade do paciente: aqui, não teria problema misturarmos dados coletados em 2007 com 2015 e, ainda, com dados de 2019 **desde que tenhamos a idade** do paciente: note que isso acontece porque os dados biológicos das pessoas em si não mudaram muito nos últimos anos, mas o fator determinante aqui é a idade (até porque certos resultados de exames podem ser bons ou ruins dependendo da idade do paciente, por exemplo, o batimento cardíaco e pressão sanguínea).

Assim, o ponto que gostaria de trazer para você é que, até o momento, o tempo era relevante de alguma forma – mas era um ator secundário. Por outro lado, existem casos nos quais o que aconteceu imediatamente antes ou depois pode ser bem relevante. Vamos trabalhar com três exemplos para melhor ilustrar isso, OK?

1.O supermercado

vamos supor que você trabalhe em um supermercado - seja como um estoquista, um caixa ou um fiscal de loja. Agora, vamos supor que eu faça a seguinte pergunta: “Quais são os dias em que o supermercado terá mais movimento?”.

a. Ora, se você tem uma certa experiência nesse local, poderá dizer algo como: “É fácil: de segunda a quinta o movimento é tranquilo. Na sexta, aumenta muito de noite. No sábado também é uma loucura. No domingo fica lotado, principalmente de manhã. Ah: também em dias de pagamento ou antes de feriado fica uma loucura aqui! Mas, olha só: para conseguir te falar certinho eu preciso saber qual é o dia do mês que você precisa saber. Como o movimento vai aumentando aos poucos, é legal saber como estava o movimento algumas horas antes”.

2. Comprando pela internet

você pensa em comprar mais memória para o seu computador, mas decidiu que importará a partir de um site da China. Como comprará de um site estrangeiro, a compra será em dólar. Você acaba de ir ao Google e viu que a cotação do dólar não está em um valor muito bom para você.

a. Então, a sua próxima pergunta é: “Quando o dólar vai chegar a um patamar bom para que eu consiga comprar a memória? Precisava que fosse rápido, mas se continuar aumentando valerá mais a pena comprar agora”.

3. Na empresa de energia elétrica

você trabalha em uma empresa de fornecimento de energia elétrica. Para que o fornecimento ocorra sem problemas (ou seja, que não falte energia para ninguém e que também não ocorra sobrecarga em nenhum lugar; ou, ainda, para que os preços sejam ajustados para cima em momentos de alta demanda e para baixo em momentos de baixa demanda) é muito importante prever o comportamento nas próximas horas e nos próximos dias.

a. Agora, pensemos com o nosso senso comum: quando o consumo de energia será maior ou menor para uma cidade inteira?

b. Os comércios e empresas de serviços operam em sua maioria em horário comercial (isto é, entre 8h e 17h, ou 9h e 19h). É nesse horário que muitas pessoas trabalham nos computadores em lojas e escritórios com luzes, aparelhos de som, ar-condicionado, elevadores e diversos aparelhos ligados.

c. Já no início da noite, há um consumo mais residencial (com lâmpadas, máquinas de lavar roupa, aquecedores a gás, televisões e computadores), mas consumindo ao todo uma quantidade menor de energia do que os comércios. Várias fábricas ainda trabalham de noite e durante a madrugada, mas não são todas.

d. E, finalmente, o horário quando há menor consumo é a madrugada.

e. Existem pequenas alterações no consumo dependendo ainda da época do ano (inverno consumindo mais do que no verão) e de feriados.

f. Por outro lado, de forma geral, o comportamento segue alguns ciclos: aumenta durante o dia, diminui durante a noite. Aumenta no inverno, diminui no verão. Aumenta em dias úteis, diminui em fins de semana e feriados.

Antes de avançarmos, respire um pouco. Entenda os três exemplos acima. Tente compreender puramente com o seu senso comum. Concorda que existem **momentos** em que o dólar, o consumo de energia ou movimento no supermercado estarão **maiores ou menores**? Concorda também que, para que possamos prever esses valores, precisaríamos também do **histórico** para que o resultado seja legal?

É esse o ponto que gostaria de trazer: existem casos em que precisamos prever um valor e dependemos também do que houve no passado (isto é, o histórico). Aqui, as instâncias passam a ter uma dependência e uma sequência. Para ficar mais claro trarei aqui dois exemplos que já vimos antes. O primeiro é o **iris**:

Comprimento da sépala (cm)	Largura da sépala (cm)	Comprimento da pétala (cm)	Largura da pétala (cm)	Espécie
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa

Concorda que tanto faz acima a ordem das instâncias? Se as reordenássemos, a situação de cada instância não mudaria: a segunda flor nada tem a ver com a primeira, a primeira nada tem a ver com a terceira e a terceira nada tem a ver com a segunda flor. Vimos no passado que um algoritmo de ML aprenderá a definir o tipo de uma flor a partir das características daquela flor em específico. Logo, uma de fato é **independente** da outra.

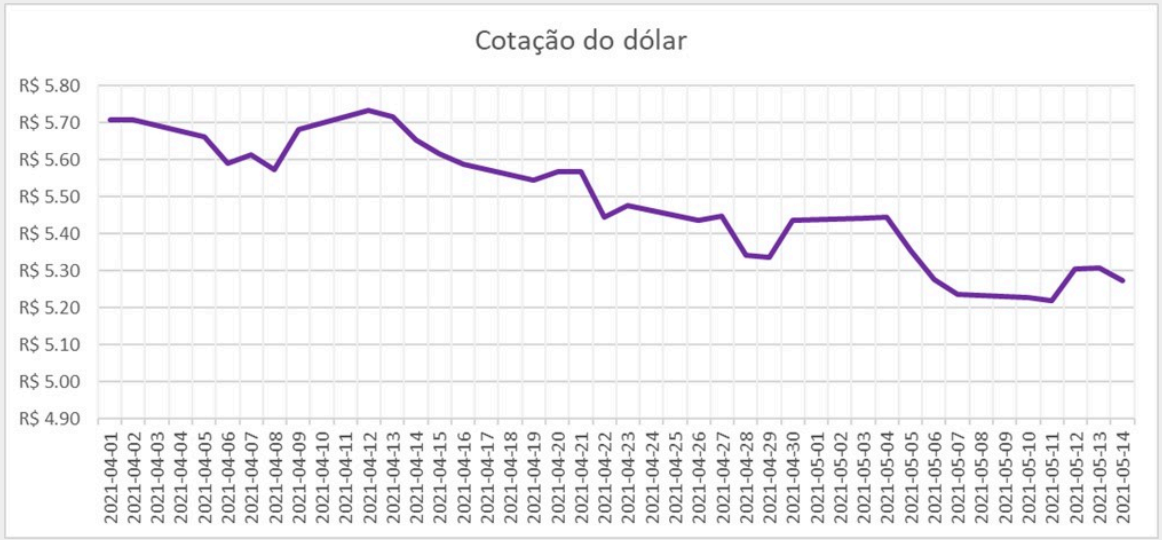
Agora, vamos à cotação do dólar:

Data	Cotacao

14.05.2021	5.273
13.05.2021	5.309
12.05.2021	5.306
11.05.2021	5.221
10.05.2021	5.227

Se queremos prever a cotação até o início do mês de junho de 2021, vamos querer saber do histórico: a tendência é que o dólar esteja aumentando ou esteja caindo durante o mês? Como estava na última semana? Como estava no dia anterior? Para pegarmos essas informações, **precisaríamos olhar para as outras instâncias**. E precisaríamos também de mais dados históricos. Observe o gráfico abaixo contendo o histórico desde o início de abril. De forma geral, a cotação está aumentando ou diminuindo? Se não ocorresse algo extraordinário, chegaríamos a quanto, mais ou menos? R$ 6.00? R$ 0.50? R$ 5.00?

#ParaTodosVerem



Fonte: Autor 2021

Ora, dias depois, a cotação **seguiu essa tendência** e alcançou R$ 5,07 em 2 de junho, e R$ 5,03 em 8 de junho. Note que para que um algoritmo pudesse prever algo parecido, ele precisaria necessariamente também de um histórico – da mesma forma que precisamos olhar o histórico segundo a imagem anterior.

Pensemos sobre o *dataset*: observe que aqui precisamos, obrigatoriamente, seguir uma sequência. As instâncias aqui passam a se chamar de **observações** (porque é como se estivéssemos tirando uma fotografia ou fazendo uma observação do que aconteceu em um determinado momento). Além disso, passaremos a adotar algumas nomenclaturas específicas para falarmos de séries temporais:

- A **frequência** é a unidade de medida dos nossos dados: estamos coletando dados a cada hora? A cada dia? A cada mês? A cada segundo? Essas unidades de tempo são as nossas frequências, e elas mudam dependendo do *dataset*: se queremos prever a temperatura de amanhã precisaríamos ter um histórico diário: um histórico mês a mês não ajudaria muito, e um histórico a cada milissegundo também acabaria tendo dados além do que precisaríamos, não acha?
- Uma observação **t** é uma que ocorreu em um determinado momento. Vamos dizer que o t (ou seja, o momento de referência) é o dia 2021-08-01 (1 de agosto de 2021; note que padronizamos sempre com o ano vindo primeiro, seguido pelo mês e, por último, com o dia);
- Assim, uma observação **t-1**é aquela que aconteceu um período antes do momento t. No nosso exemplo anterior:
    1. O t-1 seria 2021-07-31 (um dia antes de 2021-08-01).
    2. O t-2 seria 2021-07-30.
- O t-30 seria 2021-07-02, e assim por diante.Da mesma forma, uma observação **t+1**seria uma observação que aconteceu um período à frente do t. Assim, pensando no nosso exemplo:
    1. O “t+1” seria 2021-08-02 (um dia depois de 2021-08-01);
    2. O “t+2” seria 2021-08-03;
    3. O “t+30” seria 2021-08-31, e assim por diante.

Sabemos que o nome desta disciplina é Técnicas de Machine Learning, logo, há a expectativa de termos um foco em algoritmos de ML com a intenção de prever comportamentos futuros. Por outro lado, é importante que você também saiba que um dos trabalhos importantes relacionados às séries temporais envolve a **análise de séries temporais**, ou ***time series analysis*** (TSA). Uma das principais TSAs envolve o que chamamos de **decomposição de séries temporais**. Esse trabalho de decomposição separa alguns elementos-base de toda série temporal:

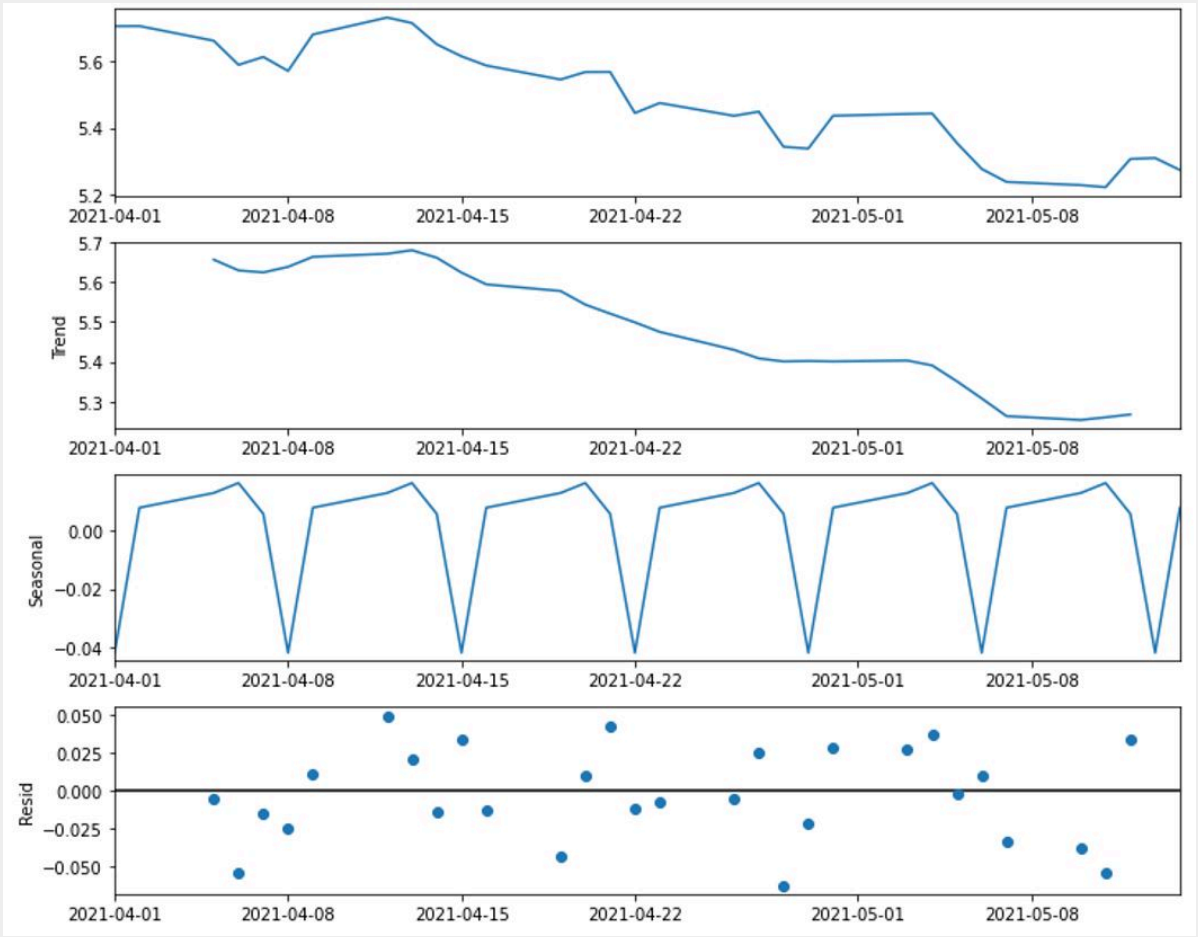
---

1. **Nível** (*level*): qual é a ordem de grandeza dos dados? Estamos falando que esse dado (cotação do dólar, quantidade de pessoas no supermercado, consumo de energia em MW) é na casa das unidades? Centenas? Milhares? Milhões?
2. **Tendência** (*trend*): de forma geral, os números estão crescendo ou diminuindo no **longo prazo**?
3. **Sazonalidade** (*seasonality*): provavelmente esse é um dos pontos mais importantes para nós. Os números aumentam ou diminuem de forma cíclica? Por cíclico, imagine assim: **toda sexta** os números aumentam muito ou caem muito? *Toda* madrugada os números aumentam muito ou caem muito? **Todo começo** de mês os números aumentam muito ou caem muito?

4. **Ruído** (*noise*): aqui ficam os valores que não são explicados pelos três itens acima. Sabe aquele chiado que às vezes se escuta em um rádio, ou uma interferência que se percebe às vezes com aparelhos eletrônicos? Imagine o ruído como sendo esse chiado ou esta interferência que não saberíamos explicar facilmente como ela surge (ou como ela desaparece).

Usemos como exemplo a mesma cotação do dólar que vimos acima. Agora, observe como fica a decomposição dela. O primeiro gráfico é a série temporal antes da decomposição e é, para todos os efeitos, o mesmo gráfico que já vimos anteriormente. Depois disso vem a ***trend***, seguida pela ***seasonality*** (*seasonal*) e, por último, o ***noise*** (*residual* ou *resid*). Note que há uma tendência geral de queda. Além disso, há uma sazonalidade detectada nas quintas-feiras de queda no dólar (ao menos observando esse breve histórico, somente). Finalmente, existem alguns ruídos cuja origem o algoritmo não conseguiu detectar.

#ParaTodosVerem



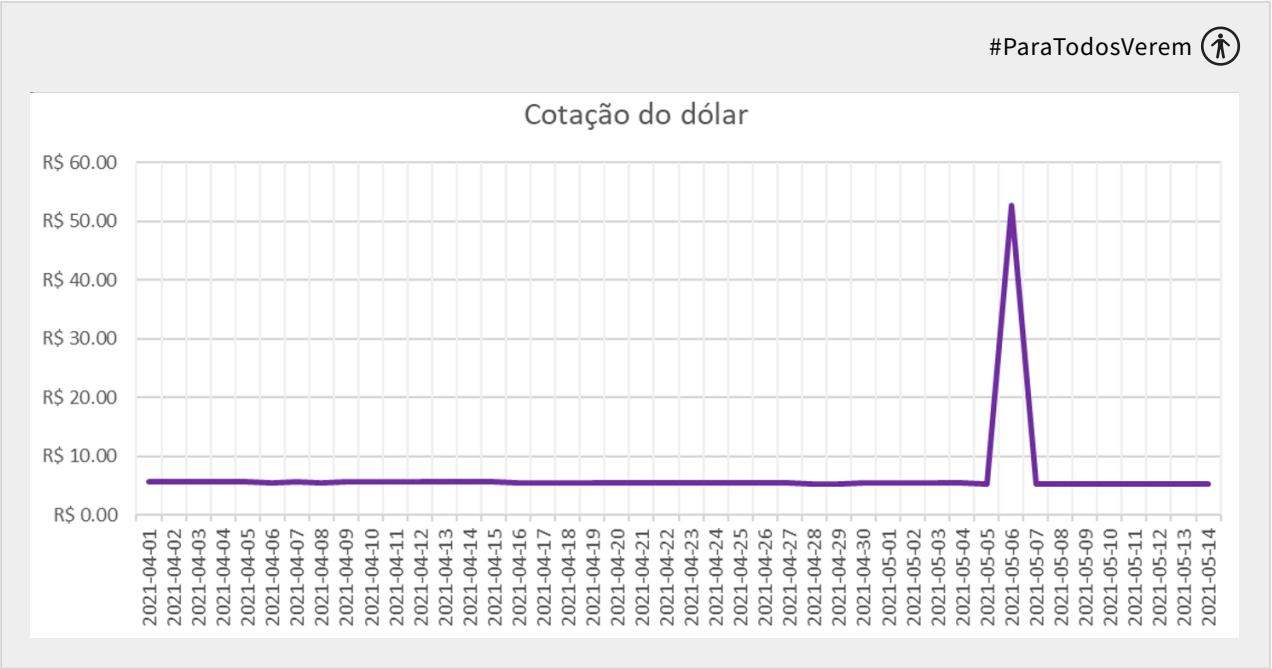
Fonte: Autor, 2021

Essa decomposição é feita com uma biblioteca bem legal chamada **statsmodels**. Ela é poderosa para análises estatísticas e para séries temporais. Para gerar a imagem acima, utilizamos uma função chamada **seasonal_decompose**.

LAVANDO A ROUPA SUJA DO PASSADO

Existem algumas preocupações relacionadas ao uso de séries temporais como, por exemplo:

1. ***Outliers***: alguns valores fora do comum (ou seja, muito acima ou muito abaixo da média) podem necessitar de um tratamento especial se forem um erro ou um evento que não queremos prever. Pensemos no dólar: observe o gráfico abaixo e os valores apresentados. Pelo seu bom senso, você acredita que isso foi um erro de digitação ou algo que realmente possa ter acontecido?

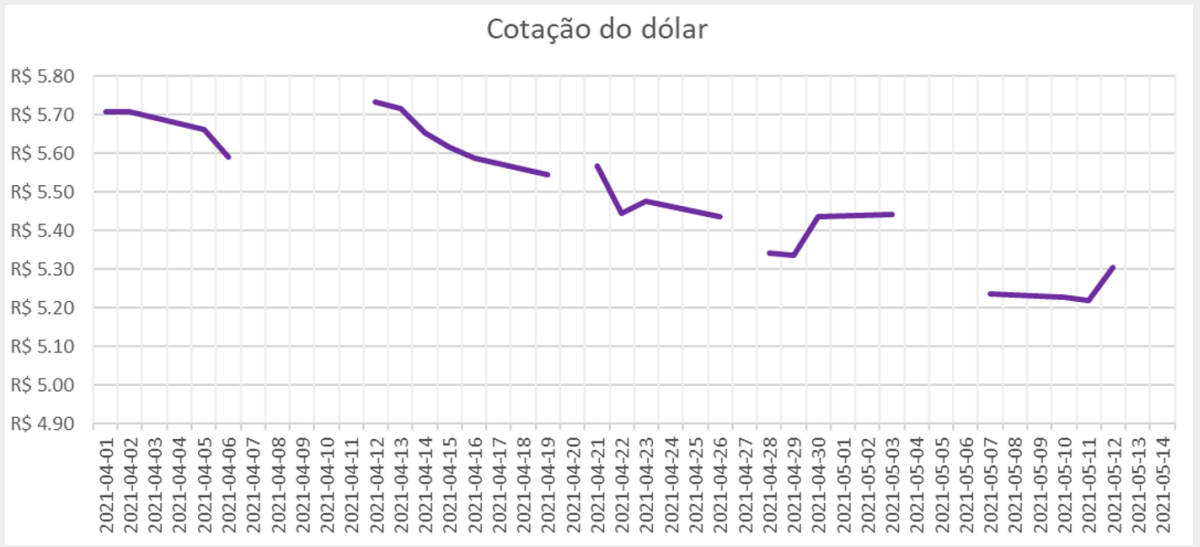


Fonte: o autor, 2021

No dia 2021-06-05 (vamos começar a usar esta notação com mais frequência: primeiro é o ano, depois o mês e, por último, o dia) a cotação ficou em 52,76. Sabemos que o dólar flutua aos poucos para cima e para baixo, mas seria bem difícil ocorrer um pico desses e normalizar logo no dia seguinte. Portanto, seria um possível erro de digitação (5,276 em vez de 52,76) que precisaria ser corrigido para que um algoritmo não acabe aprendendo que há algo muito especial na primeira semana de maio para causar isso, entende? Para ajudar a detectar esses *outliers* podemos empregar técnicas visuais como gráficos do Seaborn (como o ***boxplot***) ou, ainda, técnicas estatísticas como o ***describe*** do pandas. O ***describe*** nos informará dados como a média e o desvio-padrão. Com esses dois dados, conseguimos também entender quais são os *outliers* na nossa base de dados.

2. **Dados faltantes**: alguns algoritmos não funcionam muito bem com “buracos” no *dataset*. Voltemos ao exemplo do dólar:

#ParaTodosVerem



Fonte: Autor, 2021

Como o algoritmo vai aprender as tendências se não há uma **continuidade** aqui? Sendo assim, em alguns casos, seria interessante imputar os dados para essas lacunas (isto é, inserir um valor estimado para preencher os buracos na base histórica). Preencher com zeros pode mais atrapalhar do que ajudar (imagine o algoritmo prevendo que existem dias que a cotação fica entre R$ 5 e R$ 6 e, dias depois, fica zerada – isso não seria muito útil para nós). Preencher com a média por si só também poderia não ajudar muito. Se não tivermos acesso aos dados em alguma outra fonte, outras técnicas, como a média móvel, poderiam apoiar nesse quesito.

SLIDING WINDOW E BACKTEST

Para falarmos da criação de algoritmos preditivos, é importante introduzir dois conceitos a você: ***sliding window*** e ***backtest***.

Para que um algoritmo possa prever corretamente o futuro, ele precisará, obrigatoriamente, de dados do passado a depender da sazonalidade. Da mesma forma que pensamos na cotação do dólar ou na previsão do tempo, quanto mais para o futuro tentamos olhar, mais incerto ele é.

Dito isso, dependendo da técnica de ML aplicada, poderemos utilizar o histórico como atributo adicional. Já vimos isso brevemente na penúltima unidade. Voltemos ao nosso exemplo do dólar, mas agora com mais dados:

Data	Cotacao
14.05.2021	5.273
13.05.2021	5.309
12.05.2021	5.306
11.05.2021	5.221
10.05.2021	5.227
07.05.2021	5.237
06.05.2021	5.276
05.05.2021	5.354
04.05.2021	5.444

O ***sliding window*** (ou *rolling window*, ou *moving window*) é uma técnica que serve para reorganizar o nosso *dataset* de séries temporais para um problema de regressão. Nela, criamos uma coluna adicional com a observação t-1; uma segunda coluna adicional com a observação t-2; uma terceira coluna adicional com a observação t-3, e assim sucessivamente. A intenção disso é provermos, dentro de uma instância, todos os dados históricos para que possamos empregá-los numa técnica de regressão. O **tamanho da janela** (isto é, a quantidade de atrasos/observações) depende de cada caso. Pensemos no dólar: observando aquela decomposição de séries temporais, a tendência e o que vemos de notícias, você acha que seria tão importante assim termos um *sliding window* de 1 dia? 7 dias? 30 dias? 365 dias? Olhando o gráfico da decomposição eu sugeriria começar com algo entre 7 dias (porque há uma sazonalidade semanal) e aumentando até cerca de 30 dias (até porque mais do que isso, aparentemente, não é tão relevante, considerando todo o mercado financeiro, condições políticas e demais fatores). Para meteorologia, às vezes um *window* maior ou menor será interessante. Logo, isso depende de cada caso e depende, sobretudo, da experimentação. Para fins **puramente** de visualização, imaginemos um *window* de 2 dias para a cotação do dólar:

Data	Cotacao	Cotacao_-1	Cotacao_-2
14.05.2021	5.273	5.309	5.306
13.05.2021	5.309	5.306	5.221

12.05.2021	5.306	5.221	5.227
11.05.2021	5.221	5.227	5.237
10.05.2021	5.227	5.237	5.276
07.05.2021	5.237	5.276	5.354
06.05.2021	5.276	5.354	5.444
05.05.2021	5.354	5.444	
04.05.2021	5.444		

Note algumas coisas:

1. O valor do **Cotacao_-1** nada mais é do que o que aconteceu no dia imediatamente anterior. Já o valor do **Cotacao_-2** é o que ocorreu dois dias antes.
2. Perceba que há um buraco entre os dias 7 e 10 de maio. **Nesse caso específico**, não há problemas: dias 8 e 9 foram finais de semana e sabemos que a cotação do dólar não mudaria nesses dias, e o que ocorre na segunda é uma continuação direta do que houve na sexta, e não do domingo. Agora, se esse buraco estivesse em um dia da semana, o qual não fosse também um feriado, precisaríamos tratar de alguma forma esse valor – seja inserindo manualmente, preenchendo com uma média móvel ou qualquer outro tratamento.
3. Perceba que o dia 5 de maio não possui nada na **Cotacao_-2** (porque o mínimo que temos no histórico é o dia 4 , portanto, não existe nada antes do dia 4). Perceba também que o dia 4 de maio não possui nada na **Cotacao_-1** e na **Cotacao_-2** pelo mesmo motivo: não temos o histórico dos dias anteriores.

O ***backtest*** é uma forma de testarmos a *performance* de um algoritmo para séries temporais. Lembra do train_test_split que vimos anteriormente? Quando falamos de séries temporais não é muito legal dividirmos nosso *dataset* de forma aleatória como fizemos até o momento ao trabalharmos com os problemas de regressão e classificação que vimos. Voltemos ao caso do dólar: vamos supor que estamos no dia 14 de maio e queremos saber se ele funcionará bem para o futuro. Ora, como naturalmente não possuímos os dados do futuro, precisaríamos avaliar a *performance* do algoritmo de outra forma – ou seja, o quão bem ele está acertando as previsões. O *backtest* então é uma forma de dividir a nossa base **pelo tempo**: utilizamos as *n* últimas observações para o teste e as demais para o treino. No caso do dólar, poderia ser algo como “Vou usar o histórico de 2021-01-01 a 2021-04-30 para o treinamento, e 2021-05-01 a 2021-05-14 para o teste”. O algoritmo **não sabe** o que ocorreu em maio, mas nós sabemos: dessa forma, conseguiremos comparar as previsões do algoritmo para maio com o que de fato ocorreu. Isao pode ser um bom ponto de partida para entendermos se o algoritmo está ou não gerando boas predições.

Vamos comentar sobre três formas de trabalharmos com séries temporais:

1. Tratando como um problema de **regressão** e utilizando algoritmos de ML para tal.

CRIANDO ALGORITMOS PREDITIVOS DE SÉRIES TEMPORAIS

2. Utilizando uma técnica estatística chamada **ARIMA**.
3. Utilizando uma técnica de ML chamada **Prophet**.

Essas são algumas das principais formas de se trabalhar com séries temporais – logo, a intenção é que você saiba como cada uma delas funciona. Para demonstrar isso, usaremos o mesmo exemplo da cotação do dólar como ponto de partida – agora com mais datas:

Data	Cotacao
14.05.2021	5.273
13.05.2021	5.309
12.05.2021	5.306
11.05.2021	5.221
10.05.2021	5.227
07.05.2021	5.237
06.05.2021	5.276
05.05.2021	5.354
04.05.2021	5.444
03.05.2021	5.442
30.04.2021	5.437
29.04.2021	5.338
28.04.2021	5.343
27.04.2021	5.449

26.04.2021	5.436

IMPORTANTE

Atenção: trouxemos o exemplo do dólar por ser algo mais fácil de compreender e trabalhar no contexto da disciplina. Por outro lado, aplicações financeiras provavelmente funcionarão melhor com uma boa dose de econometria, uma área de estudos que envolve muita estatística. Na prática, é bem difícil de prevermos preços e cotações futuras – se fosse fácil, conseguiríamos enriquecer com Bitcoin em rapidamente. Por outro lado, é um caso real para testarmos no contexto da disciplina com a intenção tão somente focada no processo de aprendizagem.

Regressão

O passo a passo para se trabalhar com um problema de regressão inclui:

1. Adoção de um *sliding window*.
2. Remoção das instâncias com o *sliding window* incompleto.
3. Opcionalmente, a inclusão de algumas informações de data que podem ajudar na previsão (neste caso, o dia da semana pode ser um bom indicador – podemos adotar que a segunda seria o código 0, a terça seria o código 1, a quarta seria o 2, a quinta o 3, a sexta o 4, o sábado o 5 e o domingo, o 6);
4. Remoção da coluna de data.

Usando como base o *dataset* da cotação de dólar teríamos como resultado do passo 1 o seguinte:

Data	Cotacao	Cotacao_-1	Cotacao_-2	Cotacao_-3	Cotacao_-4	Cotacao_-5	Cotacao_-6	Cotacao_-7
14.05.2021	5.273	5.309	5.306	5.221	5.227	5.237	5.276	5.354
13.05.2021	5.309	5.306	5.221	5.227	5.237	5.276	5.354	5.444
12.05.2021	5.306	5.221	5.227	5.237	5.276	5.354	5.444	5.442
11.05.2021	5.221	5.227	5.237	5.276	5.354	5.444	5.442	5.437

10.05.2021	5.227	5.237	5.276	5.354	5.444	5.442	5.437	5.338
07.05.2021	5.237	5.276	5.354	5.444	5.442	5.437	5.338	5.343
06.05.2021	5.276	5.354	5.444	5.442	5.437	5.338	5.343	5.449
05.05.2021	5.354	5.444	5.442	5.437	5.338	5.343	5.449	5.436
04.05.2021	5.444	5.442	5.437	5.338	5.343	5.449	5.436	
03.05.2021	5.442	5.437	5.338	5.343	5.449	5.436		
30.04.2021	5.437	5.338	5.343	5.449	5.436			
29.04.2021	5.338	5.343	5.449	5.436				
28.04.2021	5.343	5.449	5.436					
27.04.2021	5.449	5.436						
26.04.2021	5.436							

Após o passo 2:

Data	Cotacao	Cotacao_-1	Cotacao_-2	Cotacao_-3	Cotacao_-4	Cotacao_-5	Cotacao_-6	Cotacao_-7
14.05.2021	5.273	5.309	5.306	5.221	5.227	5.237	5.276	5.354
13.05.2021	5.309	5.306	5.221	5.227	5.237	5.276	5.354	5.444
12.05.2021	5.306	5.221	5.227	5.237	5.276	5.354	5.444	5.442
11.05.2021	5.221	5.227	5.237	5.276	5.354	5.444	5.442	5.437
10.05.2021	5.227	5.237	5.276	5.354	5.444	5.442	5.437	5.338
07.05.2021	5.237	5.276	5.354	5.444	5.442	5.437	5.338	5.343

06.05.2021	5.276	5.354	5.444	5.442	5.437	5.338	5.343	5.449
05.05.2021	5.354	5.444	5.442	5.437	5.338	5.343	5.449	5.436

Após o passo 3:

Data	Cotacao	Cotacao_-1	Cotacao_-2	Cotacao_-3	Cotacao_-4	Cotacao_-5	Cotacao_-6	Cotacao_-7	DiaDaSemana
14.05.2021	5.273	5.309	5.306	5.221	5.227	5.237	5.276	5.354	5
13.05.2021	5.309	5.306	5.221	5.227	5.237	5.276	5.354	5.444	4
12.05.2021	5.306	5.221	5.227	5.237	5.276	5.354	5.444	5.442	3
11.05.2021	5.221	5.227	5.237	5.276	5.354	5.444	5.442	5.437	2
10.05.2021	5.227	5.237	5.276	5.354	5.444	5.442	5.437	5.338	1
07.05.2021	5.237	5.276	5.354	5.444	5.442	5.437	5.338	5.343	5
06.05.2021	5.276	5.354	5.444	5.442	5.437	5.338	5.343	5.449	4
05.05.2021	5.354	5.444	5.442	5.437	5.338	5.343	5.449	5.436	3

Após o passo 4:

Cotacao	Cotacao_-1	Cotacao_-2	Cotacao_-3	Cotacao_-4	Cotacao_-5	Cotacao_-6	Cotacao_-7	DiaDaSemana
5.273	5.309	5.306	5.221	5.227	5.237	5.276	5.354	5
5.309	5.306	5.221	5.227	5.237	5.276	5.354	5.444	4
5.306	5.221	5.227	5.237	5.276	5.354	5.444	5.442	3
5.221	5.227	5.237	5.276	5.354	5.444	5.442	5.437	2

5.227	5.237	5.276	5.354	5.444	5.442	5.437	5.338	1
5.237	5.276	5.354	5.444	5.442	5.437	5.338	5.343	5
5.276	5.354	5.444	5.442	5.437	5.338	5.343	5.449	4
5.354	5.444	5.442	5.437	5.338	5.343	5.449	5.436	3

Observe que todas as colunas são numéricas. Com isso, poderemos usar técnicas de regressão como os algoritmos do scikit-learn, XGBoost e LightGBM para prever a coluna **Cotacao**.

Prophet

O Prophet é uma biblioteca poderosa criada pelo Facebook e que funciona bem com dados com uma boa sazonalidade. Em sua forma mais básica, você somente necessita de duas colunas: uma contendo a data (e que precisa ser chamada de **ds**) e outra contendo o valor (e que precisa ser chamada de **y**). Logo, aquela base da cotação do dólar ficaria assim:

ds	y
14.05.2021	5.273
13.05.2021	5.309
12.05.2021	5.306
11.05.2021	5.221
10.05.2021	5.227
07.05.2021	5.237
06.05.2021	5.276
05.05.2021	5.354
04.05.2021	5.444

03.05.2021	5.442
30.04.2021	5.437
29.04.2021	5.338
28.04.2021	5.343
27.04.2021	5.449
26.04.2021	5.436

ARIMA

ARIMA significa ***A**utoregressive **I**ntegrated **M**oving **A**verage*. É um modelo estatístico que possui três características principais:

1. *Autoregressive* (AR): essa característica trata da relação de dependência de uma observação com as últimas observações (isto é, as observações t-1, t-2, t-3 e assim por diante).
2. *Integrated* (I): essa característica trata da subtração (diferenciação) das observações para que a série fique estacionária. Uma **série estacionária** é uma série sem sazonalidade e sem tendência (lembra que falamos sobre isso antes?).
3. *Moving average* (MA): essa característica trata da média móvel entre a observação t e o ruído de uma média móvel das últimas observações.

Assim, o ARIMA possui três parâmetros chamados de p, d e q: o **p**, usado pelo AR, refere-se à quantidade de termos autorregressivos a serem considerados (ou observações anteriores). O **q**, usado pelo I, refere-se à quantidade de vezes que realizamos as subtrações para fazer com que nossa série temporal fique estacionária. O **q**, usado pelo MA, refere-se ao tamanho do *sliding window* para calcular a média móvel.

Em python, temos algumas opções:

1. O ARIMA, disponível pelo statsmodels, no qual você necessita passar os parâmetros p, d e q anteriormente;
2. E o AutoARIMA, disponível pela biblioteca pmdarima, o qual calcula automática os parâmetros p, d e q para você.

SAIBA MAIS

Para definir os melhores parâmetros para um modelo ARIMA, é necessária uma análise do comportamento da sua série temporal

utilizando alguns critérios estatísticos. A nossa intenção é a de que você saiba que existe esta técnica, mas que acabe focando o **desenvolvimento** de técnicas de ML como o Prophet e algoritmos de regressão.

Por outro lado, caso tenha interesse em se aprofundar no uso do ARIMA encorajamos que teste diferentes parâmetros de **p**, **d** e **q**. Como conteúdo complementar, o Dr. Jason Brownlee escreveu um ótimo tutorial em seu site.

MÃO NA MASSA

Note que existe um *notebook* em python na VM intitulado ***Semana5_SeriesTemporais***. Esse *notebook* mostra diferentes implementações para um *dataset* expandido com as cotações do dólar utilizando as técnicas que comentamos anteriormente.

Observe as diferentes formas nas quais houve o treinamento e o teste. Observe também o ***backtest*****,** o ***sliding window*** e o preenchimento com a **média móvel** em uso, na prática.

Para ajudar na construção do seu entendimento, assista ao vídeo **É hora do show!** em que demonstramos o processo de treino e teste de um modelo de ML.

VIDEOAULA: Aí tá chovendo? Aqui tá chovendo

Uma das utilizações clássicas de séries temporais é na área da meteorologia – tanto para prever a temperatura dos próximos dias como também para prever a chuva. Isso possui implicações em áreas como agropecuária e logística. Vamos ver como criamos um algoritmo preditivo utilizando uma base histórica de chuvas para prever o que acontecerá no futuro?

CONCLUSÃO

Nesta unidade, nos aprofundamos no emprego de séries temporais. Existem diferentes formas de realizarmos o treinamento de séries temporais – aqui, mostramos como podemos trabalhar como um algoritmo de regressão, com o uso do Prophet e com o uso do ARIMA. Também introduzimos conceitos como o *backtest* e *sliding window*.

REFERÊNCIAS BIBLIOGRÁFICAS

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

LENZ, M. L. *et al*. Fundamentos de aprendizagem de máquina. Porto Alegre: Sagah, 2020.

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