Fala turma!

Eu estou com o Guilherme Silveira, cofundador da Alura, e agora a gente vai fazer

uma aula de Machine Learning. E o Gui me falou antes de a gente começar,

quando a gente estava conversando aqui, que essa é uma das aulas mais populares quando uma empresa contrata a Alura

para fazer essa aula, para dar um bootstrap na empresa,

para dar uma inicialização, e bater a vontade

de todo mundo começar a usar Machine Learning dentro da empresa. Então, o material que a gente

vai ver aqui é bastante legal, bastante rico, e eu considero isso

um belo privilégio aqui do canal. Então, Gui...

Tudo bom, meu caro? - Tudo bom?

Beleza, Filipe? Vamos lá! Posso sair mandando aqui? - Mão na massa! - Já vou criar... - Isso é uma coisa que

eu tenho curiosidade, Gui... Machine Learning,

para quem está de fora, e eu imagino que seja a mesma coisa para muitas pessoas estão

vendo o vídeo também... Dá uma assustada do tipo: "Poxa, a gente vai conseguir fazer

Machine Learning de verdade aqui dentro deste vídeo hoje?" - A gente consegue. E primeiro com alguns exemplos

de dados que eu vou criar para que a gente tenha uma simulação,

um exemplo que funcione... E depois, com outros exemplos

de dados que vão se assemelhar mais com a realidade de empresas. Mas é claro, a gente não vai estar

implementando a matemática por trás do aprendizado de máquina. Então, se a gente quiser,

a gente pode estudar isso também. Sem problemas. Não tem problema nenhum

aprender a implementar rede neural. Super possível. A gente tem curso inclusive disso.

Não tem problema nenhum... Mas, realmente, para a gente ver
em 15 minutos a coisa funcionando e implementar juntos, aí a gente utiliza bibliotecas
que fazem isso para a gente. - Show! - Beleza?
- Então, partiu! - Vamos lá! Então, para começar, a gente fecha aqui e dá um "No, thanks"
para o Google Colab e põe um nome nisso aqui. Então, sei lá...
"Introdução a Machine Learning". Espero que eu possa aumentar
essa fonte um pouquinho... E vamos lá. Então, vou pegar
um exemplo que a gente conversou lá para trás num
dos primeiros vídeos desta playlist em que a gente falava: "Olha... Como uma criança aprende
se um animal é um porco um cachorro?" Lembra que eu fiz umas perguntas maldosas? - Com certeza.
- Do assunto? Então, a ideia é essa.
Eu vou tentar representar um porco e um cachorro com algumas características. E ver se um algoritmo
é capaz de aprender o que é um porco e o que é um cachorro. Então, por exemplo,
eu vou pegar um porco, o "porco1", e vou defini-lo como sendo
três características diferentes... Se ele tem pelo longo... Então, a primeira característica
é se ele tem pelo longo. Então é 1, se ele tem pelo longo. É 0 se ele não tem pelo longo. A segunda característica
é se ele tem perna curta. 1, se ele tem perna curta...
0, não tem perna curta. E a terceira característica
é se ele faz "au au". Se ele faz "au au" é 1,
se ele não faz é 0. Três características. Super válido. Então, eu tenho um porco que,
por exemplo, não tem pelo longo, ele tem perna curta
e ele não faz "au au". Então, o meu problema,
o problema que eu vou ter vai ser... Eu tenho diversos animais, por exemplo,

um cachorro que não tem pelo longo, tem perna curta e faz "au au". Então, eu tenho vários animais e eu quero classificar esses animais

numa classe A ou B, numa classe 0 ou 1, numa classificação binária,

que se chama esse problema. Então é um problema clássico

de Machine Learning, classificação binária. Poderia ter classificação

mais do que binária, uma classificação em várias classes. Se esse e-mail é de vendas,

se esse e-mail é do comercial, se esse e-mail é jurídico,

se esse e-mail é técnico... - Talvez eu especulo que venha daí

a grande escala da Inteligência Artificial. Porque para um ser humano

fica fácil analisar o que é um porco

e um cachorro nesta dimensão aqui. Mas quando começa a misturar

muita coisa, muitas dimensões, muitas classificações,

o cérebro não começa a dar conta. É aí que a Inteligência Artificial começa

realmente mostrar sua força de alavanca? - Com certeza.

Pode ser um dos motivos, porque o relacionamento

entre essas variáveis e a classificação, o ser humano seja incapaz de perceber, então esse é um motivo... Tendo em vista que aqui

eu tenho três variáveis e eu tenho uma variável aqui proposital para que fique fácil

para a gente, ser humano, tentar identificar

se é um porco ou um cachorro. Mas no mundo real, para saber se

uma pessoa vai fraudar ou não um sistema, se é uma fraude ou não é... Para saber se uma pessoa vai pagar um débito ou não vai... Para saber se uma pessoa vai ter

dificuldade em um curso ou não vai... Eu tenho um monte de variável e não é óbvio o relacionamento

entre essas variáveis... E se a pessoa vai ou não vai

passar numa prova, sabe? Não é óbvio isso. Então, aprender esses relacionamentos é uma coisa

que o Machine Learning

é capaz de fazer que a gente pode ter certas dificuldades... Podemos ter,

não quer dizer que vamos ter... E outra é o volume mesmo, não é? Se eu tenho dez animais

para classificar, beleza... Chama um especialista de animal

aí para classificar. Agora, no momento que eu tenho

1 bilhão de imagens para classificar... Boa sorte, não é?

Não adianta sair contratando especialista que é financeiramente inviável. - Justo!

- Então são dois motivos importantes. E esse é um tipo de classificação. Tem outros problemas,

como regressão, agrupamento, diversos outros problemas

que o Machine Learning também ataca... Inteligência Artificial, etc. Vou descrever outros porquinhos aqui. Um outro porquinho

que também não tem pelo longo, tem perna curta

e é um porquinho que faz "au au". Tem porquinho que faz "au au".

Procura que você vai achar. Está aqui, entendeu?

Não tem como! É estatística! Algum porquinho

no mundo faz "au au", entendeu? - Tem muito porco, não tem como!

- Porquinho fake news. - É! Algum deles faz, entendeu? Conversa lá com ele,

faz alguma coisa que sai um "au au". [1, 1, 1]

Então, tenho três cachorros e três porcos. E o que a gente fala é:

a gente separa isso aqui e a gente quer treinar

que nem a gente treinava criança. Eu tenho uma criança... A criança olha esses seis animais... Então, esses aqui são meus animais,

o "porco1", o "porco2"... Opa! "porco2", "porco3"... "cachorro1", "cachorro2" e "cachorro3". Então, eu tenho seis animais que poderiam ser os meus itens,

seja lá o que for. É o que a gente tem para classificar. Então, como isso aqui a gente vai usar

para treinar minha criança, o meu filho, a minha filha,

ou meu algoritmo... A gente chama isso de "treino". É o meu treino. Então, esses são os meus dados de treino. Mas nos dados de treino

para treinar o meu filho e minha filha não adianta ter só

as características de um porquinho, eu tenho que dizer para o meu filho

ou para a minha filha, ou para o algoritmo, qual desses seis é porco,

e qual desses seis é cachorro. Então, eu também preciso

falar as classificações. Eu vou chamar o porco de 1

e o cachorro de 0. Então, três 1's e três 0's. Então, o "porco1" é 1... o "porco2" é 1

e o "porco3" é 1... "cachorro1" é 0,

"cachorro2" é 0 e "cachorro3" é 0. Então é comum nesses algoritmos... É uma abordagem, não significa que é a única maneira

de fazer classificação binária... É você ter aqui os seus dados de treino e as classificações que você vai usar

para esse treino. - Então, só para esclarecer...

- Pode falar... - Esse [1, ,1 , 1] e [0, 0, 0]

que está aqui embaixo em classificações eles não têm... - Nada a ver...

- Eles têm a ver com certeza, mas não têm nada a ver com

o [0, 1, 1, 0] dos dados lá de cima. - Isso, nada a ver com os de cima. Aqui é [0, 1] e significa 0 cachorro 1 porco. Assim como esse [0, 1] aqui,

não tem nada a ver com esse [0, 1] aqui. Esse [0, 1] aqui,

o terceiro [0, 1] do cachorro, quer dizer "au au", e esse [0, 1] aqui quer dizer perna curta,

não tem nada a ver um com outro, só tem a coincidência

de estar na mesma linha, portanto pertence ao mesmo animal. - São as features desse animal. Cada uma dessas colunas é uma feature, que é uma característica

que a gente tem aqui que poderia ser binária, como essas... Mas poderia ser um número maior,

até mesmo não inteiro, poderia ser o tamanho

em m² de um apartamento, 147,35m²... Poderia ser um texto,

poderia ser várias coisas. - Pode ser várias coisas...

- Interessante! - Não precisa ser 0 ou 1. Claro... Cada tipo de variável

que a gente tem aqui, que faz "au au", não sei o quê... Vai ter um tipo de tratamento diferente. Então, variáveis binárias como essa, a gente consegue direto. Uma variável categórica,

como São Paulo, Rio de Janeiro, Brasília... Que são Estados ou o Distrito Federal... Tem várias categorias. Então é um número finito de categorias. Então, tem outra maneira de trabalhar. Tudo isso a gente vê

em cursos de preparação dos dados, "preparamentos" é bizarro, né? de preparação dos dados

ou de como extrair essas features aqui, como trabalhá-las. Faz todo sentido. - Perfeito.

- Então, a gente tem essas três features, seis itens que a gente

vai classificar no treino, e eu chamei de treino em "classificacoes". Só que o que eu quero é ter uma criança,

um filho, uma filha, ou um algoritmo, que eu vou passar para esse filho,

filha ou algoritmo todos esses porcos

e todas essas classificações, e o que eu quero depois para

esse meu filho, essa minha filha, ou esse meu algoritmo

é perguntar assim... "E esse animal,

é um porco ou é um cachorro?" E aí, o meu filho ou filha fala 0 ou 1,

cachorro ou porco. Então, o que a gente quer,

no final das contas, a gente quer alguém,

um ser humano ou uma máquina, que seja uma função matemática

que recebe três variáveis... Recebe X, Y ou Z,

seja lá o que for... Esses três números 0 ou 1... E devolve ou um 0 ou um 1. Então, o que a gente está procurando aqui é uma função que mapeia isso aqui nisso aqui, que mapeia três variáveis em 1, 0 ou 1,

três variáveis em 1, 0 ou 1... Três variáveis em 1, 0 ou 1. É isso o que eu quero do meu filho. Eu mostro três variáveis para ele, se tem pelo longo,
se tem perna curta e se faz "au au"... E eu quero que meu filho diga 0 ou 1. Isso é a definição
de uma função matemática que recebe X e devolve Y. Então, o nome tradicional aqui
é de "treino x", ou em inglês... "treino x" e "treino y". Então, "X" é o que eu vou dar
para o meu filho, e "Y" eu também vou dar para
treinar o meu filho ou o meu algoritmo... Mas já vou falando
para ele assim: "Olha... Isso aqui, esse porco, essas três
variáveis aqui estão ligados com esse. Essas três com esse, essas três com esse,
essas três com esse..." Aí, o meu filho teoricamente
aprende alguma coisa. Tudo bem até aqui? - Perfeito. - Vamos escrever o código do meu filho
ou da minha filha, né? Então, vou usar uma
biblioteca que já existe, o "scikit-learn"... No scikit-learn tem uma
implementação de um algoritmo... Uma variação, tá? Tem vários algoritmos...
Vou pegar um simples de propósito, que é o "LinearSVC". O LinearSVC é um modelo. Eu vou chamá-lo aqui
e vou criar esse meu modelo... É como se fosse um cérebro vazio. Pensa num cérebro vazio
que não foi treinado ainda. E eu falo para esse modelo se adaptar, para tentar encaixar
esses dados na cabeça dele. Então, encaixa esses dados
do "treino x" e do "treino y", faz um feat desses dados
dentro do teu modelo, dentro do teu cérebro. E agora, o meu modelo está treinado. Então, eu vou rodar aqui, como a gente fez
para rodar nas outras aulas... Eu tenho aqui um modelo treinado. Ele está falando:
"Isso aqui é um LinearSVC." Beleza, agora eu vou pegar
um animal misterioso. "animal_misterioso". O meu animal misterioso tem pelo longo, tem perna curta e faz "au au". - É um cachorro. - Você acredita que é um cachorro.

Eu também acredito que é um cachorro... - Eu também acredito que é verdade.

- Vamos ver o que o... - O que o modelo vai predizer para a gente, o que ele acredita que é

esse animal misterioso. Só um cuidado aqui... O jeito que eu estou passando

é um único animal. Mas o "predict"

recebe uma lista de animais. Então, eu tenho que colocar

dentro de uma lista. É só um cuidado

para não receber um erro aqui. E ele devolve para a gente

uma lista de previsões. Como só tem um animal, - uma única previsão: 0.

- Perfeito! - Zero era... Cachorro. - Cachorro. - Acertou!

Neste caso, acertou. E a gente poderia, claro,

fazer uma série de animais. Eu poderia criar vários animais aqui,

por exemplo, "misterio1", que é [1, 1, 1]... "misterio2", que é [1, 1, 0]... E o "misterio3", que é [0, 1, 1]. E aí, a gente coloca que o "teste x"

são esses três ani... Desculpa! Aliás, eu coloquei "treino" lá ou "teste"? "Treino", está certo... Então, meu "teste" agora

é testar esses três animais. Até porque entra aquela questão

que a gente tinha falado antes. Lá naquele primeiro vídeo, eu falei... Eu acho que eu falei,

ou eu não falei? Não lembro se falei ou não. Então, falamos agora... Acho que a gente comentou sim... É muito bizarro eu treinar com um porquinho

e depois te perguntar... "Este animal, o que ele é?". Então, repara que o cachorro [1, 1, 1]

a gente já tinha treinado. Eu já tinha mostrado para o meu filho

que [1, 1, 1] era um cachorro. Se eu mostro o mesmo

cachorro para o meu filho, existe uma chance de o meu filho

simplesmente decorar... "Aquele aquele animal é um cachorro",

e não aprender. Ele não aprendeu,

ele só decorou. Então, quando você passando exatamente os mesmos valores

de "treino" para o "teste", para validar se ele adivinhou... Aqui eu estou validando se ele adivinhou com o mesmo animal que eu treinei, pode ser que o algoritmo tenha só decorado, o que não é legal. Então, a gente vai treinar com vários exemplos. Estou treinando com vários exemplos... E o meu "teste", o que eu sei de verdade é que o primeiro é um cachorro e os outros dois são porcos. Eu sei.

Então, isso aqui é o que eu sei. E agora, o que eu quero fazer é calcular todas essas previsões. "previsoes = modelo.predict(teste_x)"... E vamos ver as "previsoes". Opa! As previsões... E as previsões foram [0, 1, 0].

Quer dizer, acertou só os dois primeiros. Como ele acertou só os dois primeiros, a gente pode calcular essa taxa de acerto. Como eu falei, a gente está usando o "sklearn" e o "sklearn" vai ter tudo isso para a gente, um monte de medidas, de métricas... Como, por exemplo, a taxa de acerto. A taxa de acerto é o que a gente chama de "acurácia". É o quão certa é a acurácia. - Mas isso é uma coisa interessante da acurácia, porque eu acho que, às vezes, a expectativa que alguém tem de Machine Learning ou do computador, é colocar um peso muito maior no computador do que no próprio ser humano. Tipo... "Ah, o ser humano pode errar. É ser humano, não é? Ser humano erra... Mas computador não, computador tem que ser perfeito sempre." Isso é uma realidade? Pelo que eu estou vendo, nada a ver. - Então... Lembrando que aqui são dados criados propositalmente para a gente ter esse resultado, para a gente ter isso de exemplo. - Mas na vida real? - Na vida real, a maior parte das coisas que a gente for fazer não vai dar certo de primeira. Se dá uma taxa alta, por exemplo... Essa taxa de acerto... Você testou 100 valores. Você pegou 100 que não tinha treinado

e testou. E desses 100, você acertou 97. Aí você fala:

"Nossa! A taxa de acerto foi 97... ... no que eu nunca tinha visto antes!" Animal! Provavelmente, a gente errou

alguma coisa no caminho, porque a gente não chega

em taxa de acerto de 97%... - Sério?

- Nunca! - A regra é nunca.

- Interessante... Se chegou em 97 logo de cara, provavelmente a gente errou alguma coisa. - Existe algum número

que a pessoa possa se basear? "Putz, eu estou longe para ruim,

estou longe para bom..." - É difícil... Mas existe uma sacada assim.

Existem algumas sacadas. A gente mostra isso

um pouco mais para frente. Mas eu vou citar agora,

porque é super importante e você citou. Então, como eu meço se

o meu algoritmo foi bom ou foi ruim? Deixa eu chamar esse "accuracy_score"

só para gente ver o resultado dele. Eu quero comparar com o que eu sei,

que é o "teste_y"... As previsões. As previsões que eu tive. Eu podia imprimir isso aqui

bonitinho com porcentagem e etc... Então, ele está me mostrando

que eu acertei 66%. Duas de três: 66%. 66% é bom ou é ruim? Não tenho ideia, olhando assim... E 66% de fraude é bom ou é ruim? - Vai depender do assunto, não é? - Primeiro vai depender do assunto. Então, por exemplo,

a taxa de detecção de AIDS... 50% está bom ou está ruim? Provavelmente está ruim, porque você não quer falar para as pessoas

que elas têm uma doença que muda o comportamento, o dia a dia da pessoa

e precisa ser tratada e etc., quando ela não tem. Ou falar que ela não tem quando ela tem. Então, em uma questão de doença é super importante o falso-positivo

e o falso-negativo... Tem coisas que você

quer tomar muito cuidado e essas taxas são super importantes

que sejam altas. Em outros lugares,

imagina que o ser humano é incapaz de detectar se uma estrela

tem planeta ao redor dela a olho nu. Mas se o algoritmo, né?

A gente é incapaz, certo? É incapaz.

Se você olhar, você não é capaz, ponto. Mas se você tiver mais dados,

talvez sim... Então, a taxa de acerto nossa

é zero ou talvez 50. Sim ou não, se você chuta,

talvez fosse 50 ou algo do gênero. - Dá para chutar...

- É! Mas a taxa do computador é 55. Maravilha! Então, o que costuma

ser usado como base é... O algoritmo tem que ser melhor

do que um ser humano. - Pelo menos melhor.

- Perfeito! - Porque se ele não for

melhor que um ser humano, é lixo, não serve para nada. É claro, tem a questão do custo de novo.

Talvez seja muito custoso pedir

para um ser humano fazer essa tarefa. E aí, você pede para a máquina, mesmo que a taxa de perfeição,

ou de acerto, ou outra métrica que você esteja usando

não seja equiparável a um ser humano. - Mas, eventualmente,

é uma decisão de negócio sobe o que fazer. - Não é só técnico.

- Por um lado... Isso! Por um lado é de negócio,

que é isso o que a gente está discutindo, e por um outro lado existem algumas

medidas básicas que a gente pode usar. Se você pegar no mundo

ou na base de dados que você tem... Se você perceber que 80%

dos seus animais é cachorro e 20% é porco, o mínimo que eu espero do meu algoritmo

é que ele acerte sei lá 81%. Porque se ele acertar 60%,

ele é muito burro. Porque era muito mais fácil

eu chutar cachorro para todo mundo. Tudo bem? Então, o algoritmo chutar

cachorro para todo mundo tem uma acurácia muito melhor

do que esse algoritmo aqui. Só que também tem o falso-positivo, etc.,

que você também tem que cuidar. - É que nem olhar para

os sistemas solares que tem por aí e falar: "Aquilo lá não tem planeta." Bom, é bem possível que vai acertar, mas... - Se a maior parte não tem, você vai acertar simplesmente

porque a maior parte não tem. Então, uma métrica... Ou o inverso, se a maior parte

não tivesse você iria errar, porque a maior parte não tem. Então, isso aqui é um classificador.

Isso que eu chamei, esse LinearSVC é

literalmente um classificador. E se você for no sklearn,

ele tem um classificador burro. Eu não sei se a palavra é burro,

bobo ou algo do gênero, uma palavra provavelmente

menos ofensiva... DummyClassifier. E o DummyClassifier você pode usar,

você pode importá-lo aqui e o que ele vai fazer é... Ele vai chutar, por exemplo,

o mais frequente. Ou chutar proporcionalmente. Então, ele faz esses chutes

meio simples para a gente. Então, tecnicamente,

o mínimo que a gente espera é utilizar um classifier

ou um regressor básico bobo, como esse tem vários... Tem para classificação,

tem para regressão... Dá para fazer para time series, dá para fazer para

vários casos uns dummies... E você tem que ser melhor que isso.

Porque se você for pior que isso, não serve para muita coisa. Então, você tem que ser melhor que isso e não pode gastar muito recurso. Não adianta o meu programa

precisar de dias para ser 1% melhor de uma coisa

que 1% não traz muito resultado. Aí também não adianta muito. Então, tem todo esse balanço

dessa parte técnica que você tem que ser melhor

do que um ser humano... E de algum dummy, porque senão você usa

uma heurística simples. É suficiente para o seu programa. E por outro lado, você tem que ver se aquilo

faz sentido para a tua empresa. E a realidade é que muitas vezes

você cria esses modelos, esses estimadores, etc... E eles são piores

do que heurísticas simples. Ou, às vezes, eles empatam,

mas são mais rápidos. Às vezes, eles empatam

e são mais devagar. Então, a maior parte das vezes falha, demora para a gente chegar num resultado que é vantajoso para a empresa.

Não é comum chegar logo de cara não. - Tá. - Então, este aqui era

o primeiro exemplo do porquinho. Eu posso tocar um outro exemplo, então? Vai ser bem parecido

só para a gente mostrar isso um pouco mais

como se fosse no mundo real? - Perfeito. Vamos lá!

- Pode ser? Beleza!

Eu tenho aqui, só preciso pegar uma URI aqui que eu tenho... Eu acho que vou roubar

e pegar essa URI de lá... Não tem problema,

eu vou roubar. Lembra do "pandas"... - que lia arquivos CSV?

- Com certeza! Então é ele que a gente vai usar.

Deixa eu dar um scroll down aqui para a gente ver mais bonitinho... É ele que a gente vai usar. Eu tenho um outro Notebook desse curso que tem aqui a URI bonitinha

que eu vou utilizar para ler. Então é essa URI aqui. Essa URI, eu vou ler... "pd.read_csv(uri)"... Devolve os dados...

Esses são os meus dados e vou olhar cinco linhas. Então, esses dados são de mentira,

eu que gerei... E você tem aqui

se cada um desses usuários, cada um é um usuário que acessou o meu site. Então, este usuário aqui,

que é o "0", a primeira linha... Ele acessou a página inicial, ele acessou a página

"Como o meu site funciona", o meu produto funciona... não acessou a página de contato e não comprou o meu produto. Então, se a gente der

uma olhada nos dados... Vou pegar um sample, tá? Aqui em samples aleatórios. Então, eu tenho pessoas

que acessaram tudo e compraram, pessoas que acessaram

algumas páginas e não compraram, pessoas que acessaram

algumas páginas e comparam... Tem de tudo! Tem meio que todos os tipos de pessoas. O que eu gostaria de saber é... Se a pessoa acessou tais páginas,

será que ela vai comprar ou não? Porque de acordo com isso,

eu quero entrar naqueles... Sabe aqueles pop-ups chatos

que aparecem no meio da tela na hora que você menos quer?

Você está para clicar no link, e aparece e você clica no chat, não é? Você só quer que aquele chat apareça se a pessoa não está entendendo

o seu site e vai desistir. Você não quer atrapalhar a pessoa. Então, seria interessante

abrir o chat só nessas situações. Numa situação em que a pessoa

já está interessada no meu produto. - Que sensacional!

Estou mega empolgado com esse exemplo. - É uma simplificação.

Esse problema é muito mais complexo. Aqui eu coloquei em 0 e 1. A gente teria que colocar, por exemplo,

a ordem que a pessoa acessou as páginas. - Mas já é ótimo para entender o quão

próximo a gente está de um caso real. - Isso é muito fantástico.

- Isso! Era isso o que eu queria trazer... Mas repara que...

Poxa... A gente está igual a antes. A gente tem o X,

que são três colunas, e o Y, que é uma coluna. É o mesmo problema. Do jeito que ele está e do jeito

que as variáveis foram colocadas, é o mesmo problema. Eu posso falar assim:

O X são os dados, as colunas "home", "how_it_works", perdão o inglês... E o "contact". Perdão novamente,

que aos poucos eu assassino a língua. "x.head" para saber aqui... Então, a gente tem as três colunas. E o Y que é o segundo lado,

que é onde vai usar o "teste" e etc... Ele é o "dados"...

E chamava... Qual era mesmo? Bought! Sei lá como fala também,

vai ser "bought" agora. Então, beleza!

É bought! Beleza! Esse é o Y. Vamos ver cinco elementos

dele também para ver que está direitinho? Está direitinho. Se eu tenho X e eu tenho Y,

o que eu posso fazer? Treinar e rodar. Então, eu poderia...

Não que vá funcionar, tá? Eu poderia fazer isso aqui... Treinar. Então, eu vou copiar e colar lá embaixo,

só para a gente ir mais rapidinho... Vou copiar...

Lembra que a gente tem o X e o Y. Então, eu treinei o meu algoritmo... Eu poderia testar... Então, modelo...

Eu vou deixar na mesma célula... "modelo.predict"... Aí eu tento prever o X e essas são as minhas previsões. E aí, eu posso chamar o "accuracy_score" do Y com as minhas previsões. Isto é, eu comparo as previsões com Y. E vamos ver essa taxa de acertos.

Eu só vou multiplicar por 100 para que dê em porcentagem. Vamos ver? 96%. - Está errado. - Está errado, né?

Primeiro, porque está errado mesmo... E segundo,

porque os dados foram gerados, eu gerei esses dados

para que a gente tivesse sucesso. - Perfeito! Mas o que está errado aqui? A mesma coisa que eu discuti antes. Eu falei: "Olha... Mas se eu treinar e eu testar com os mesmos valores,

eu estou viciando o meu algoritmo." - Perfeito, perfeito! - Então, aqui eu estou treinando e testando

com os mesmos valores.

Então, o que eu tenho que fazer é... Eu tenho que de alguma maneira

separar o "treino" e o "teste". Adivinha o que existe para a gente? "train_test_split"... Você passa o X e Y e ele vai separar para a gente

o "treino_x" e o "treino_y", o "teste_x" e o "teste_y". Eu espero que seja essa a ordem, porque eu sempre erro

essa ordem desses caras. "treino_x"... Acho que eu errei.

Acho que é "teste_x"... Estou dando uma roubadinha

aqui na minha cola, tá? "treino_y"...

A ordem das variáveis é infernal aqui. - Mas isso é uma coisa que a gente

não aprendeu no Python ainda. Tu estás fazendo

o resultado dessa função... - "train_test_split" vai ser...

- Aaahhh... Boa! - Boa pergunta...

- Vai voltar um array? - O que é isso?

- Boa pergunta! Isso mesmo! Então, a gente pode criar uma função. Eu posso fazer uma função

que retorna cinco números. E ela simplesmente

dá um "return 1, 2, 3, 4. 5". Então, quando eu estou

retornando esses cinco números, quando eu vou chamá-la, eu posso fazer

"a, b, c, d, e = retorna_5_numeros". E aí, ele já mapeia.

Várias linguagens não deixam fazer isso. Muitas linguagens não deixam. Vou imprimir o "a" aqui para a gente ver... E vou imprimir o "e", que é o 5. Então, é verdade.

Este cara aqui devolve diversos valores, esses quatro valores... E aí, eu já atribuí

a quatro variáveis distintas. - Perfeito, perfeito.

- Beleza. Então, essa célula eu vou apagar

para não ficar no meio... Mas é isso mesmo. Então, eu estou devolvendo

as quatro arrays. Se a gente olhar... Bom, aqui eu tenho que mudar agora. Isso aqui é antes de

treinar. Antes de treinar... Então, eu treino com o "treino_x" e "treino_y"... E eu testo com o "teste_x" e comparo com o "teste_y". - Uma última pergunta antes de rodar...

Esse "train_test_split", essa função... Ela é global?

De onde ela está vindo? - Ela não está vindo de lugar nenhum.

Não ia funcionar, né?! "sklearn train_test_split"... Tem aqui

"model_selection.train_test_split". A gente tem que importá-la. Obrigado! Ia dar erro mesmo. "from sklearn.model_selection

import train_test_split" Verdade! Sempre tem que importar de algum lugar. Básico assim. Toda essa estrutura base

vai estar no sklearn mesmo. Vou rodar e... 92%! Piorou, mas é um pouquinho mais realista. Pelo menos,

a gente separou os dados de verdade. Se você olhar, por exemplo,

o "treino_x", você vê que ele tem 74 colunas agora. Então, a gente pode imprimir

até mais bonitinho... Eu acho que tem o shape direto... 74x3.

74 colunas... Desculpa, linhas...

Por três colunas. E o "teste_x" tem... 25 linhas por três colunas. Então, por padrão,

o "train_test_split" separa 25%. Se você olhar aqui,

ele vai ter "test_size"... Por padrão, 25%. Então, essa é uma maneira de a gente

quebrar e validar o nosso modelo. Tem outras maneiras mais complexas, mas a gente leva um curso inteiro

para falar das mil maneiras que tem, literalmente, de validar. - Pode falar...

- E com isso, eu consigo entender ou até induzir um caminho mais feliz

para o meu usuário? Por exemplo,

eu consigo extrair essa informação? Tipo...

Das pessoas que foram na "home", foram na... Desculpa, eu esqueci o nome

das outras páginas, mas... - "Como funciona" e "contato"...

- "Como funciona" e depois "contato"... E isso é o caminho da compra, por exemplo.

Eu consigo extrair essa informação? - Percebe que aqui na maneira

que eu modelei, que eu extraí as features,

eu não extraí a ordem. Então, eu não sei se a pessoa

foi primeiro para a "home", - depois pro "contato"...

- Ou em qualquer ordem. - Isso, aqui está qualquer ordem. Então, a partir daqui, eu poderia tentar

induzir conclusões do gênero: como eu acredito que

essa pessoa não vai comprar, aí vem a pergunta:

o que eu posso fazer com isso? Então, tem duas coisas

que a gente poderia fazer com isso. Uma é tentar calcular a correlação

de acesso a essas páginas com as vendas... E aí, a gente poderia

acreditar erroneamente que a correlação é causalidade... E por isso, pessoas que acessaram

a página de contato tendo acessado as outras duas, compram.

Isso é uma correlação... A gente acredita que elas compram, então eu forço as pessoas

a entrarem na página de contato. É aquilo que eu falei,

correlação não implica causalidade. Então, não dá para ter certeza disso. Tem que fazer uns testes

bem complexos para fazer isso. A outra maneira de utilizar

esse modelo do jeito que ele está, a maneira adequada,

seria a seguinte... Então, beleza... Eu percebi que

esse modelo é capaz de prever se uma pessoa vai ou não vai comprar,

com X% de acerto. Dado que este modelo está simples

e os dados foram criados para isso. O que eu posso fazer é... Então, uma pessoa acessou o meu

site

e está numa página X. Eu calculo para ela a previsão,

eu faço esse predict para ela... Alguns modelos,

tipo o LinearSVC... Deixa eu ver aqui se o LinearSVC tem,

eu não lembro... O sklearn, o LinearSVC... Eu acho que ele não tem probabilidade... Não, não tem. Alguns deles tem uma probabilidade.

Então, ele te fala ainda: "Olha, eu acredito que é 77%

a probabilidade de comprar." Então, você pode usar um threshold

um pouco diferente aí. - Legal! Dado isso,

qual experimento eu poderia fazer? Então, eu coloco

um experimento em produção. Se a pessoa acessar as páginas

e eu julgar que ela não vai comprar, para metade das pessoas,

eu mostro um formulário de chat... Para outra metade, eu não mostro. Então, eu estou fazendo um teste,

um randomized... Control group...

Eu nunca lembro a frase certa... Randomized controlled trial

é isso o que eu estou fazendo. Um teste controlado aleatório,

algo do gênero. Então, eu estou dividindo

os meus usuários em dois grupos. E para alguns deles,

dando um comportamento do meu site, e para outros,

dando outro comportamento. Como se fosse um Teste A/B. E aí, eu eu vejo se tem diferença

no resultado desses dois. Se tiver, então legal...

Aí, eu posso jogar fora o teste e fico com essa feature. Se o resultado for que quem eu abri o chat deu a mesma coisa de vendas no meu site, então joga fora a feature,

porque não está servindo para nada. - Engraçado, porque isso tudo

era feito de forma manual no passado. Não exclui ainda o tato do ser humano, mas numa escala gigantesca com

milhões de acessos e milhões de páginas, chega uma hora que não dá,

tem que tem que usar isso. E quem usar isso,

vai sair na frente com certeza. - Com certeza! Pega o caso do Netflix

com as automatizações deles. Eles automatizam todo esse processo. Então, ele vai automatizar e gerar as thumbnails... Então, ele gera a thumbnail do filme baseado em dez imagens e dez fontes e formas de colocar o título ali na thumbnail. Então, ele faz essa mistura, coloca tudo, tenta calcular o que está acontecendo, e à medida que vai vendo o que vai acontecendo no experimento, vai tomando a decisão de qual thumbnail vai ser melhor para uma pessoa ou para um tipo de pessoas, ou outro tipo de pessoas. - E do jeito que a gente está fazendo aqui, são comandos manuais. Então, obviamente, deve ter uma forma de isso ficar sempre em formato de sistema vamos dizer assim, sem ficar se retroalimentando e melhorando os seus resultados sem ninguém tocar, correto? - Com certeza. Esse modelo que estamos usando, que é o LinearSVC, que tem diversas limitações...

Primeiro, ele é um modelo linear, então ele só é capaz de aprender comportamentos lineares... Então, à medida que a gente estuda Machine Learning, etc, a gente vai ver que...

Deixa eu jogar esse cara aqui para baixo... Não é para baixo, é para cima...

Para baixo, pronto! A gente vai ver que, na próxima aula, por exemplo, deste curso, a gente vê um conjunto de dados cujo comportamento não é linear. Aqui são as pessoas que venderam um produto e que não venderam. Então, não é uma linha.

Isso aqui é uma curva. Então, essa curva aqui que é de terceiro grau ou alguma coisa do gênero, o LinearSVC não é capaz de aprender isso. Um algoritmo, um estimador linear não é capaz. Eu até tenho um pouquinho mais para frente, eu acho que tenho esse gráfico do LinearSVC... A gente faz isso durante o curso. Este aqui é o LinearSVC

aprendendo. Essa é a reta que ele traça. E esses pontos roxos e os amarelos
que estão aqui são os que fazem diferença. Quer dizer,
ele não aprendeu porcaria nenhuma. Não aprendeu nada. E aí, você vai otimizando
e usando algoritmos não lineares para aprender isso. Então, voltando lá para a sua pergunta... Primeiro, a gente tem que ir vendo
outros tipos de estimadores para ser capaz de melhorar
o nosso algoritmo. Então, o LinearSVC é bem a casquinha... E esses tipos de modelo
que eu estou mostrando são modelos que você treina
e depois usa. Então, você não vai ficar
treinando toda hora. Você não treina toda hora... - Entrou um usuário novo, você "retreina".
- Ah entendi! - Esse algoritmo não costuma
funcionar assim. Esses algoritmos. - Fica "cozinhado" ali o comportamento.
- Isso! Você roda o modelo,
você treina o modelo, viu que vale a pena "deployar" esse,
você "deploya"... E você fica usando-o com vários predicts. Daqui a um mês, você roda de novo e gera um modelo novo,
ou sei lá quanto tempo. É claro, você pode automatizar
esse processo para rodar toda semana, toda noite,
seja lá quando for... Mas existe outra categoria
de algoritmos de Machine Learning e de Inteligência Artificial, que são os "Reinforcement Learning". Os de reinforcement learning são live. Então, à medida que você
está jogando coisas ali, ele vai aprendendo e já está adaptando o modelo dele,
sendo capaz de prever coisas baseadas naquilo
que ele acabou de aprender. Então, esses não precisam desse tempo,
dessas duas fases separadas em geral. - Que sensacional! - E é uma outra área do Machine Learning,

literalmente... Com suas vantagens e desvantagens

e crescendo cada vez mais hoje em dia. Beleza... Então, acho que ele é um pouco disso.

Com isso, realmente a gente vê... O basiquinho, só o básico é... Mas o básico é estrutura de muita coisa,

pelo menos de classificação... Separar os nossos dados

em conjuntos menores para evitar o vício. Então, tenho que separar em treino e teste. Como eu separo em treino e teste?

A gente viu uma maneira. A gente na Alura tem

um curso inteiro de dez maneiras e existem várias outras

maneiras de separar, cada uma com sua importância,

cada uma tem seu sentido. Aqui, por exemplo, se eu rodar de novo...

Rodei de novo... 96%, como assim? Então, existem algumas coisas

que dá para melhorar aqui. Isso aqui ainda está sendo

influenciado por aleatoriedade, a gente provavelmente não quer isso... Então, tem várias coisas

que dá para a gente melhorar que a gente vai vendo durante o curso. A primeira aula é aquela que

a gente falou do porco e do cachorro, a segunda, a gente fala disso

e da aleatoriedade... Depois, a gente discute

essa questão da linearidade, a gente discute a questão das dimensões... Eu citei para você sobre...

Aqui, a gente tem três dimensões no X. Visualizar três dimensões,

a gente até consegue, né? Mesmo um plot,

a gente conseguiria na visualização. Mas depois colorir para sinalizar

a quarta dimensão aqui, começa a ficar mais complicado. E isso porque a gente tem quatro. Com cinco dimensões,

com 15 dimensões, com 25... Super difícil. Então, a gente tem um curso

que toca esse assunto, que fala de redução de dimensionalidade. Como é que eu pego um

conjunto de dados
com uma dimensionalidade enorme, seja para visualização,
seja para treinar os meus algoritmos. Porque quanto mais
coluninhas a gente tem aqui, mais diversas complicações surgem, tanto de lentidão quanto de pequenos desvios errados,
de vícios que aparecem. Então, a gente discute isso aqui e discute mais a fundo
em outros cursos de novo, que a gente está aqui introduzindo. E depois, a gente vai
para outros tipos de algoritmos, como árvore de decisão, que geram coisas mais interpretáveis. Alguns algoritmos de Machine Learning
são difíceis de interpretar a olho nu. Aqui na árvore de decisão,
muitas vezes ela é interpretável. Então, posso ver aqui que,
por exemplo, se o preço de um carro for menor que R$60 mil, então vem para cá. Se o preço de um carro
é menor que R$40 mil, então vem para cá. O carro será vendido no meu site. Se o preço do carro for maior que R$40 mil, aí ele analisa aqui uma outra diferença.
Até R$40 mil vai ser vendido. Então, na verdade, ele está falando
que menor que R$40 mil vai ser vendido. Porém, se tiver entre R$60 mil
e R$240 mil, ele vem para cá. Se for menor que R$100 mil,
não vai ser vendido. Então, você começa a analisar
de acordo com o preço... Ah desculpa, aqui é quilometragem. E a quilometragem se o seu carro
vai ou não vai ser vendido no site. Então, será que um produto
que colocou no meu site e o preço que a pessoa
está querendo vender... Imagina que eu estou querendo
vender o meu carro por R$100 mil... Será que vai ser vendido
ou é melhor eu avisar ao usuário? "Olha, R$100 mil é muito. Eu acho que ninguém vai comprar
o teu carro por R$100 mil. Você não quer colocar mais barato?" Então, essa árvore aqui consegue

justificar para a gente porque ela acha que

a classificação é uma ou é outra. E hoje em dia, com todas as discussões

válidas sobre os preconceitos e os bias que os algoritmos podem ter e têm, é muito importante que a gente

entenda por trás o que está acontecendo para que a gente

tente evitar essas situações. - Perfeito!

É tipo uma forma de um ser humano, comparado a uma Inteligência Artificial que tem a sua alimentação

de processamento de dados... Então é uma forma de o ser humano

conseguir entender o output desse Machine Learning. - Pois é!

- Muito interessante... Porque existem... Imagina que existe

aquele caso lá clássico dos Estados Unidos, esqueci o nome do projeto que julga se uma pessoa que está sendo julgada

no tribunal por um crime pequeno, eu não sei dizer exatamente o caso,

por um crime pequeno... Se ela vai ser reincidente ou não. E de acordo com isso,

o que o algoritmo disser, a pessoa vai para cadeia ou não vai. Ou ela fica solta, faz serviço comunitário,

ou alguma outra coisa. Eu com certeza estou

falando isso de maneira errada. Tem detalhe melhores. Mas é algo do gênero. E é óbvio, encontraram bias e preconceitos

de raça e etc., nesse algoritmo. Então é super delicado e é importante a gente entender

o que está acontecendo de errado para a gente tirar

esses bias desses algoritmos. - Muito bom. Muito bom! Então, acho que com isso aqui

dá para ter uma ideia. Isso são algoritmos de classificação. Mas da mesma maneira que

a gente classificou entre 0 ou 1, e que se exploda o que é 0 ou 1... Pode ser spam ou não spam...

Pode ser cachorro ou porco,

pode ser vende ou não vende... Pode ser fraude ou não fraude... Poderia ser 0, 1, 2,

poderia ser 0, 1, 2, 3, 15... Poderia ser descobrir o preço

de um apartamento entre 0 e 1 milhão... Poderia ser... Quais são os grupos de usuários

que vão passar numa prova versus...

Aí é classificação de novo. Poderia ser agrupar usuários do meu site

por tipos de gostos musicais... Então, poderiam ser diversos

tipos de conclusões. Todos eles vão seguir meio que

esse tipo de padrão de execução. - Por isso que o pessoal

está com muita sede por ter dados, por ter uma base de dados, porque dá para treinar melhor

esses modelos, correto? - Com certeza!

- Porque virou uma febre virou uma febre ter os dados.

Todo mundo tinha dados antes, mas agora... Ficou há um bom tempo já

uma febre de um jeito diferente. - Com certeza!

É fundamental ter os dados e... E mesmo assim, a gente não tem

garantia de que vai sair alguma coisa. Quantas vezes a gente tem aqui

dentro da Alura mesmo... Quantas vezes,

nos últimos cinco anos, eu testei tantos algoritmos e não consegui. De novo,

não porque o algoritmo é ruim. Por dois motivos... 1. Não sou o maior expert da área. Tem gente que manja muito mais

do que eu de tudo no mundo inteiro... Mas o principal motivo é porque é difícil. Não é fácil fazer a coisa funcionar. Você precisa ter muitos dados que tenham

aquela informação que você precisa... E de todos os projetos que já testei, o primeiro assim de grande sucesso

foi o "recomendador". A gente tem um recomendador de curso

dentro da Alura. Então, à medida que você

vai fazendo cursos, a gente vai recomendando

os próximos cursos para você. - Massa! - E esse a gente conseguiu
algum resultado interessante de que a pessoa faz mais cursos,
termina mais cursos por ela estar utilizando esse recomendador. Mas foi o primeiro que realmente trouxe...
A gente olha, as pessoas estão fazendo mais cursos, e está trazendo mais resultados
tanto para os usuários quanto para a gente. Mas de novo,
isso depois de falhar várias vezes. Em várias perguntas que a gente queria
responder e não conseguiu responder. - Eu acho que é até normal
como em qualquer área também, principalmente em uma área
super nova como essa. - Com certeza. Mas é um pouco diferente
da Engenharia de Software talvez... Talvez, talvez eu esteja errado. Mas, em geral, quando a gente

vai desenvolver um software, a gente já meio que fala:
"Beleza! É possível." O Machine Learning,
por exemplo, eu gostaria de saber
antes de começar a criar um curso... Isto é, antes de um professor
ou uma professora investir horas e horas e dias
de tempo na criação de um curso, eu gostaria de saber se aquele curso
vai ser um sucesso ou não. Não financeiro,
mas de qualidade, né? Porque eu vou investir
um tempão e dinheiro para criar um curso que
a qualidade é ruim? É super ruim isso... E tentamos já de tudo que é jeito,
e por enquanto nada. A gente não conseguiu encontrar
um modelo que descreva isso. Só umas coisas muito óbvias. Um curso avançado
e que tem menos de 1h de vídeo... Provavelmente, vai dar super errado. Então, algumas coisas
muito óbvias aparecem. Mas... De resto, não conseguimos. Não temos dados que

descrevam isso para a gente. - Então, show Gui! Agora que a gente
teve uma introdução de Machine Learning, qual é o próximo passo
que uma pessoa pode dar, a pessoa que mergulhou nesse assunto
de Inteligência Artificial? O que falta para ela colocar no seu leque? - Legal!
Então, se a gente for lá na Alura, na área de cursos de dados,
de Data Science, de dados em geral... A gente acessa dados em geral... Você vai ver lá na parte de
Data Science que a gente tem também
Data Visualization. Então, a gente viu Machine Learning,
a gente falou um pouco de Machine Learning, falamos de Data Science
e tem Data Visualization. De novo, lembrando... Dentro de Machine Learning
tem várias coisas, várias que a gente não falou,
por exemplo, processamento de textos, em descrever,
criar resumos automáticos de textos, categorias de textos... E Data Science
fala bastante de estatística... Em Data Visualization, como a gente visualizar esses resultados,
mostrar resultados e de maneira a não induzir mentiras, provavelmente,
a gente não quer induzir mentiras... E a contar histórias que fazem sentido
de uma maneira fácil de serem compreendidas. Então, existe uma área de estudo,
que é a Data Visualization, que é super legal e é importante a gente ter
algumas noções também. Pelo menos umas noções básicas aqui. Depois, à medida que você
quiser se aprofundar, você se aprofunda, claro... - Show, então é a próxima
aula dessa playlist. E é a última aula dessa playlist... Então, vamos lá? - Beleza! - Então, show!
Para você conferir a aula
de visualização de dados, é só clicar aqui. Fechado? Valeu!