02

A regressão logística

Transcrição

Seguem os dados do filme Zootopia:

```
movieId,Titulo,Documentary,Sci-Fi,Mystery,Horror,Romance,Thriller,Crime,Fantasy,Comedy,Animatio
9999999,Zootopia,0,0,0,0,0,0,0,1,1,1,1,0,1,110,27.74456356,?????
```

[00:00] Agora nós já demos uma entendida em como que estão nossos dados e tudo mais, vamos aplicar o modelo que resolve isso. Aqui está justamente tudo aquilo que nós escrevemos até então.

[00:15] Lembrando das aulas anteriores, nós vamos usar um modelo pra resolver esse tipo de problema que, apesar do mesmo nome e apesar até mesmo da forma de importar ser extremamente parecida, e o nome, na verdade, não é o mesmo, é parecido, ele resolve outro tipo de problema, que é a regressão logística. Que apesar de ter de ter esse nome, regressão, ele não faz uma regressão, ele tenta calcular e resultar pra você alguma coisa entre 1 e 0, e fazer essa classificação que nós queremos.

[00:48] Se nós dermos uma olhada no tipo de dado que nós temos, nós temos ali nosso dado de treino, nós vamos ver que ele é um dataframe lá do Pandas. Fica muito mais fácil de nós trabalharmos com o scikit-learn com todos os nossos dados sendo arrays, array de dados, mais por uma convenção, uma facilidade também.

[01:07] No primeiro caso que nós vamos fazer, nós vamos simplesmente transformar tudo o que nós temos, ou seja, o nosso treino, vou estar apertando seta pra cima, o nosso treino, o nosso teste, o nosso teste labels e o nosso treino labels. Nós vamos transformar tudo isso daqui em array.

[01:27] No primeiro caso o que nós precisamos fazer? Eu vou escrever tudo aqui de uma vez, depois eu copio e colo lá, pra nós darmos uma acompanhada.

[01:36] Eu quero que o meu treino, eu vou ter que importar o numpy, então estou aqui importando o numpy. E eu quero o “np.array”, eu quero que os meus dados de treino recebam “np.array”, o próprio treino, que é esse dataframe, o numpy entende isso, eu preciso reformular esses dados pra ser o tamanho dos meus dados de treino. No caso, o próprio treino e tenha o número de colunas que eu passei.

[02:20] E a mesma coisa para o teste, eu quero que o teste receba “np.array”, passando “(teste).reshape” o próprio tamanho do teste e o número de colunas que eu passei. Se nós viermos aqui e copiarmos todas essas coisas. Fiz isso daqui, que agora é o meu type, meu type é um numpy array.

[02:46] Se eu quiser ver então o que isso daqui é, olha só, é justamente a mesma coisa que nós tivemos lá da primeira vez, dos filmes, todo o meu conjunto de filmes, então se eu der um “treino.shape” devo ver isso, 508 filmes e 15.

[03:03] Pra nós vermos que bateu, vamos dar uma olhada no nosso data frame de labels? 508, também pra uma única coluna só. No teste aconteceu exatamente a mesma coisa, eu tenho teste shape e o tipo dele, numpy array também.

[03:24] E agora como é que nós fazemos? Nós vamos ter que fazer um pouquinho diferente pra nós trabalharmos com os nossos labels. Nós fazemos o “treino_labels”, nós vamos fazer de um jeito diferente, mas é pra nós conseguirmos acabar aprendendo coisas novas, labels. Nós estamos pegando os valores desse dataframe e isso vai nos retornar um array de

arrays, um vetor de vetores. Vou até mostrar aqui pra ficar um pouco mais visível. Eu vou fazer só aqui, “Command + V”, olha só, é um array de vetorzinhos.

[03:57] Nós não queremos esse vetorzinhos, nós queremos tudo junto, tudo num caso só. E pra isso o próprio Python tem pra nós o ravel. Então aqui nós temos o ravel, que faz essa função. Se nós chegarmos aqui, “Command + V”, fez isso daqui, vamos ver agora como é que está? “treino_labels”, agora está tudo junto como nós queremos.

[04:23] Nós precisamos fazer exatamente a mesma coisa pra teste, labels recebe “teste_labels.values.ravel”. “Cltr + L”, “Command + V”. E agora nós temos o nosso “teste_labels”, e manteve tamanho. Se nós formos ver, tinha que dar 170, pro caso de teste, 170.

[04:55] Agora o que nós precisamos fazer? Nós já temos os dados assim separados da forma como nós queremos, nós precisamos criar o nosso modelo pra fazer a nossa previsão. Então, como eu falei, a forma de nós importarmos esse módulo é muito parecida com quando nós trabalhamos com regressão linear, que o linear model. Então nós temos “from sklearn. Linearmodel import LogisticRegression”.

[05:37] Então nós importamos o modelo, vamos ver só se foi ok. Agora nós precisamos falar, o nosso modelo recebe “LogisticRegression”. Nós temos o “modelo.fit”, o nosso “treino” e o nosso “treino_labels”.

[06:02] Nós fazemos o fit desse cara, nós já estávamos acostumados do próprio modelo da SKLearn, fez o fit, modelo is not defined porque nós não o copiamos aqui, fizemos isso agora, criou a nossa regressão logística. E nós precisamos simplesmente fazer o nosso conjunto de previsões. Vou chamar de “previsoes”, recebe “modelo.predict” usado de teste. Agora nós vamos aqui, “modelo.predict”, dados de teste, vamos ver essas previsões? Essas daqui foram as previsões que nós tivemos.

[06:40] Agora vamos dar uma olhada pra ver como foi esse modelo no geral, pra ver se ele foi bom, porque vai que ele chutou tudo aleatório. Depois nós entendemos muito bem o que está acontecendo ali por debaixo dos panos.

[06:55] Se nós dermos uma olhada aqui, nós vamos usar a biblioteca do sklearn, “from sklearn.metrics import accuracy_score”, então nós já temos as previsões. Nós temos a nossa acurácia, o nome é exatamente esse, em inglês, accuracy, “accuracy_store” passando os nossos “teste labels”, que são os nossos dados verdadeiros e as previsões que nós tivemos. E nós vamos descobrir qual que foi a acurácia do nosso modelo.

[07:35] Dou um “Ctrl + L” pra limpar, limpei, accuracy is not defined justamente também porque eu não importei tudo. Então “Command + C”, “Command + V”, importei. Qual que é a nossa acurácia? Ali, 75% mais ou menos para os casos de teste. Foi ok, foi uma acurácia razoável. E agora o que nós queremos entender? Vamos entender, na verdade, por que isso daqui funciona, porque funcionou, na verdade. Teve um resultado acima de 60%, foi bom.

[08:05] O que a regressão logística faz? Por que ela se assemelha muito com a regressão linear? Porque ela tenta encontrar uma reta que separe os dados, a melhor curva de reta. Só que a diferença é um pequeno detalhe, na verdade, que tem essa diferença, porque pro resultado final dessa curva ou na verdade a equação final, que forma a curva da regressão e é isso que nós vamos ver, é a função sigmóide.

[08:34] Se nós dermos uma pesquisada no Google sobre o que é essa tal função sigmóide, o nome é até estranho, um nome meio feio, na verdade, mas nós vemos que ela não é tão assustadora assim quando nós entendemos, então nós estamos abrindo aqui o Google pra dar uma olhada nisso.

[08:46] Vamos dar uma pesquisada aqui, “sigmoid function”, em inglês mesmo, função sigmóide. Vamos olhar as imagens. Essa é a equação da função sigmóide. Então no final da nossa regressão logística, ela tenta se acertar nessa equação.

[09:12] Mas você não precisa ter muito medo dessa fórmula, não. Na verdade, o que você precisa abstrair disso daqui é porque ela tenta justamente apertar nossos dados de acordo com essa curva, então o resultado final da nossa equação vai ser alguma coisa que está entre 0 e 1. Então se nós conseguimos que a probabilidade do fator ocorrer, é justamente isso que ela quer dizer.

[09:38] Quando eu falo probabilidade, eu quero dizer assim, ela vai retornar um número, e com base nesse número ela vai falar, esse número está mais perto de 1 ou está mais perto de 0. Com base nessa resposta ela vai falar, por exemplo, se nós voltarmos e subirmos um pouco, pra esse cara ele está mais perto de 1, então a chance maior dele é ser 1. Esse cara está mais perto do 0, a chance maior dele é ser 0, justamente como é o resultado dessa fórmula aqui.

[10:03] E se ela retorna justamente essa probabilidade, como é que o modelo funciona? Ele traça um limiar, um threshold, em meio, e se o valor é menor do que meio, logo ele é 0, se o valor é maior do que meio, logo ele é um. O modelo funciona, isso é interessante, na verdade, e até diferente. É outra abordagem, outra característica pra outro tipo de dado que também nós estamos trabalhando.

[10:39] Nós conhecemos, inclusive de outro curso aqui da Alura, outro classificador pra aprendizagem supervisionada, que é o Naive Bayes. Vamos comparar os dois pra ver o que acontece? Pra ver como que ele se comporta, no caso aqui, o Naive Bayes?

[10:52] Nós temos o “from sklearn”, nós importamos o “MultinomialNB”, é “sklearn.naive_bayes”. Então eu tenho o “modelo_NB”, ele recebe “MultinomialNB” e “modelo_NB.fit”. Eu quero passar pra ele então os dados de treino que eu tinha e o treino labels que era outra resposta desses caras.

[11:38] E nós falamos que os nossos modelos NB estão aqui, então agora nós temos as nossas previsões, exatamente como nós fizemos antes. Ele vai receber o “modelo_NB.predict(teste)”, enter pra nós nos guiarmos e eu vou fazer a minha acurácia.

[12:06] Vou copiar daqui, eu tenho o teste labels e previsões NB. Copiei aqui, colei aqui. Então eu tenho isso daqui, copiei esses caras. Vamos ver agora como é que se deu? Limpei a tela com “Ctrl + L”, “Command + V”, printei, qual que foi minha acurácia agora? 68%. Então fui pior. Pro primeiro caso que nós vimos, a acurácia da regressão logística foi melhor.

[12:38] E por que isso acontece? Aí que é interessante nós entendermos um pouquinho mais a fundo esses modelos. O Naive Bayes em geral lida com problemas de contagem, então nós estamos fazendo uma contagem de frequência e tudo mais, e ele calcula a probabilidade em cima disso.

[12:57] Então o Multinomial se dá melhor nesse tipo de caso, e aqui nós estamos trabalhando com um modelo um pouco mais complexo, nós não temos bem uma contagem definida de caras que aconteceram, caras que não aconteceram, nós até temos, mas nós temos mais variáveis categóricas aqui, então com base no meu vetor de características, eu quero encontrar uma curva que divida ou separe bem o meu espaço, que separe bem o meu conjunto de filmes, entre gostou ou não gostou, no caso.

[13:24] Uma coisa também que é sempre bom nós deixarmos válido, na verdade, pra problemas reais de Machine Learning, é porque aqui, em todos esses casos, eu sempre fiz essa divisão e todos esses testes pra definir a acurácia do modelo, sempre em cima dividido em treino e teste. Na verdade nós temos um caso a mais de validação, nós temos geralmente treino, teste e validação.

[13:46] E pra eu poder comparar os dois, ou seja, essa acurácia com essa acurácia pra definir qual modelo eu vou usar, o ideal é justamente separar esse modelo em validação, e então, por exemplo, eu pego meus conjuntos de teste que eu tenho, que nós separamos o teste, quebro-o na metade e treino em cima desses caras da metade.

[14:09] Esse seria o nosso caso de validação, porque numa situação real nós não temos nossos casos de teste, a resposta deles, nós não podemos treinar em cima de teste, então nós podemos estar enviesando esse cara, nós estamos dando uma roubada. E como o foco dessa aula é mais as técnicas, nós estamos dando essa roubada permitida assim.

[14:30] Nós vimos aqui a diferença, que eu tenho os modelos, em alguns casos, justamente entender o modelo um pouco mais a fundo, é um caso bom, um caso o ideal pra nós podermos fazer a nossa previsão, pra nós não sairmos fazendo besteira. Como nós vimos, nesse caso, vou até calcular tudo aqui de novo, nós termos escolhido a regressão logística é o nosso modelo ideal pra nós trabalharmos. Eu tenho aqui nossa nova acurácia, que é a mesma lá do 74% que nós estávamos mexendo da primeira vez.

[15:01] Agora nós estamos no nosso caso de teste real, vamos ver como é que o nosso modelo se comporta com o Zootopia? Se eu não me engano, acho que está em outra informação, vamos até descer tudo isso só pra ter certeza, mas não estão aqui as informações de Zootopia. Nós temos essas informações em um dataset aqui. Zootopia data, se eu não me engano é esse daqui.

[15:24] Vamos ver com esses dados de teste. Olha só, nós carregamos. São justamente essas mesmas informações, a diferença é que ao invés de bilheteria, nós temos o gosto. Nós temos o movie ID, temos o Zootopia, na verdade esse é um caso até mesmo mais simples. Então aqui o outro, que ele tem os dados mais completos. Aqui está o nosso caso.

[15:44] Nós temos exatamente todas essas informações, eu vou copiar isso e colar aqui, então vamos chamar de "Zootopia", poderia fazer a leitura, mas acredito que assim seja muito mais fácil.

[16:00] Nós temos esse array de dados, que foi a primeira transformação que nós fizemos, eu estou copiando na mão só pra ficar mais rápido de fazer, então é basicamente cada um dos tipos dos filme, as categorias e quais delas Zootopia é, se é animação, se é terror, drama, etc. E nós também não estamos interessados no ID do filme, por isso que eu não copiei. E nós temos o Zootopia data recebe isso, pra nós vermos que isso daqui está ok, vamos copiar. É por isso que eu vou apagar esse cara, pra ver se isso daqui está ok eu tenho que dar 15 no nosso shape.

[16:38] Nós vamos aqui, colamos isso, Zootopia. Se nós dermos um len nós vamos ver que deu 15. Se eu der um len em "treino [0]" deu 15 também. Então olha só, a mesma coisa, mesmo formato dos dados.

[17:02] Agora o que nós precisamos fazer? Nós vamos fazer simplesmente um predict. Nós já temos nosso modelo ali que nós copiamos de novo, nós temos um "modelo.predict(zootopia)", ou seja, nosso usuário final vai gostar ou não vai gostar de Zootopia. E ele deu um erro aqui, porque nós temos que passar isso como um array, dou um "Ctrl + L" pra limpar, limpou. Olha só, 1, ou seja, o nosso usuário vai gostar de Zootopia.

[17:32] Então nós voltamos pro nosso chefe, bem felizes, podemos chegar pra ele e falar: "chefe, o pessoal gostou, pode investir, Zootopia é a chave pro sucesso". Então é isso.

[17:47] Na verdade nós vimos agora todo o processo, a comparação entre dois modelos de Machine Learning, a escolha pra um caso ideal e como, na verdade, um modelo diferente daquilo que nós estamos acostumados também se aplica.

[18:06] Uma coisa muito legal de Machine Learning, em geral, é essa existência de diferentes modelos, diferentes algoritmos que nós temos para resolver o mesmo trabalho. E o ideal é nós conhecermos diferentes técnicas, diferentes abordagens, como elas funcionam pra determinado modelo, pra dependendo de como nossos dados estão sendo elaborados, nós sermos capazes de escolher o melhor modelo. E conhecendo mais nós acabamos fazendo e tendo um desempenho melhor.