05

A regressão linear

Transcrição

Seguem também os dados do dataset Zootopia:

```
movieId,Titulo,Investimento,Bilheteria
999999,Zootopia,27.74456356,?????
```

[00:00] Oi, pessoal. Tudo bom? Vamos dar uma revisada naquilo que acabamos de fazer. Nós importamos a biblioteca de análise de dados, o pandas. Nós importamos o matplotlib para poder desenhar os gráficos. Nós fizemos a leitura do csv. Separamos os dados nos dados de investimento, uma coluna do nosso data frame e nos dados de bilheteria. E depois desenhamos esse gráfico. Depois nós resolvemos repetir o processo, pegando uma amostra e novamente fazendo as mesmas coisas. Printando o gráfico.

[00:36] Para nós não esquecermos daquilo que fizemos, vou até entrar aqui de novo. Entrei aqui no terminal, abri o terminal. Eu estou indo lá para minha pasta. André, machine learning, curso. Entrei aqui na minha pasta. Eu posso entrar aqui no python e ficar fazendo as interações aqui dentro do python. Ou eu posso salvar tudo isso daqui num único arquivo e executar ele tudo de uma vez.

[01:03] O problema de eu fazer isso é porque eu teria que executar todo o processo e depois eu só me preocupo com o resultado final. Mas eu poderia muito bem chegar aqui hoje e falar: “python exploração.py”. Por exemplo, aqui ele está executando todos os negócios e ele gerou o gráfico da primeira vez de todos os dados, executou aqui de novo e executou esses dados para nós da amostra.

[01:30] Vou voltar aqui, mas na verdade não é isso que nós queremos. Nós queremos fazer um processo um pouco mais interativo nesse curso. Nós importamos aqui de novo os dados. Volto para o terminal e colo as bibliotecas. Ok. E nós vamos fazer novamente a leitura dos arquivos.

[01:50] Vamos estar interessados, agora, de voltar de onde partimos. Já pegando as amostras e olhando para o gráfico. Estou copiando os gráficos no command + C ou copy. E aqui colando os dados “plt.show()”.

[02:06] O que nós podemos ver a partir desse gráfico? Relembrando aqui que vimos no último vídeo. Conforme eu aumento o meu investimento, o número de pessoas também tende a aumentar. Esses dados tem uma associação positiva, uma correlação positiva. E aqui nós conseguimos ver que eles têm uma associação linear, eles formam quase uma linha. Estão meio espalhados, mas formam quase uma linha.

[02:30] E o que nós podemos pensar? Quanto mais eu aumento o meu investimento, mais número de pessoas tende a ir. Então, se eu investir dez reais e vão cinco pessoas. Se eu investir vinte reais, agora vão dez.

[02:55] E se eu aumentar e investir mil reais? Mil reais, vão umas quinhentas pessoas. Agora, um milhão de reais. Vão umas trezentas mil pessoas. E o que podemos falar para o nosso chefe? Quando mais você investir, mais gente você vai ter. Ele vai falar: legal, mas eu preciso de um dado um pouquinho mais preciso. Uma métrica um pouco mais bacana, um número mais preciso para eu ter uma noção se vale a pena ou não investir.

[03:22] E nós pensamos: como eu posso fazer isso? Eu vou ter um ponto que é formado por uma combinação de dois elementos. No caso, é um caso mais independente, que nós já temos todas as informações, que é a nossa variável de

investimento. E uma resposta para esse cara, que é a nossa variável independente, que é a bilheteria e o número de pessoas que vão.

[03:48] No caso, vamos supor, de Zootopia. Nós temos a informação do dado que queremos prever. Olhei aqui no curso, estou aqui no meu data set. Eu tenho "zootopia_data". O que tem aqui? Esses dados estão aqui. O que nós podemos ver? Vou até abrir ele aqui no Atom direto, que ele talvez fique um pouco mais bonito para nós. Eu abri ele aqui, eu tenho o meu data set, zootopia, data. Olha só.

[04:21] Aqui é o movie ID, o título. O investimento dele. No caso de Zootopia, é aproximadamente trinta milhões, mas a bilheteria nós não sabemos. Como é que conseguimos prever o que nós queremos prever? Nós temos um ponto X do investimento e nós vamos ter uma bilheteria num ponto Y associado.

[04:46] Qual a primeira ideia que vem? Vamos olhar para esse gráfico, ele forma uma linha. Se nós criarmos uma reta. Para essa reta, teremos um X e um Y associados. A variável resposta que nós queremos, que é a bilheteria e o número de pessoas que vão.

[05:09] Para ilustrar esse processo, eu vou abrir aqui o Geo Gebra. O que é o Geo Gebra? O Geo Gebra é um software de geometria, para conseguirmos enxergar um pouco melhor esses dados. Vamos entrar no Geo Gebra para explicar um pouco melhor aquilo que eu estou querendo falar.

[05:24] Eu entrei aqui, inicializei o Geo Gebra. Qual o ponto que nós queremos? Nós temos um filme que está aqui. Lembra daquele gráfico de ponto que acabamos de ver? Lembra desse gráfico aqui? Nós temos vários pontos aqui, e esses pontos estão espalhados. E qual é a ideia? É nós encontrarmos, talvez, uma reta. Se nós lembrarmos do ensino médio, nós queremos encontrar a variável resposta, o nosso Y, que é esse ponto aqui.

[05:50] Ela é formada por essa equação aqui "mx + b". Nós queremos encontrar o B e o M para que esse X esteja correspondente àquela reta. Vamos colocar os nomes. Eu tenho aqui um 2x mais sete, não menos sete. Isso aqui é uma reta. Então, vou ter um ponto X e um Y associado. Isso aqui seria uma reta de previsão. Vou ter um X, uma previsão.

[06:21] Se nós pegarmos, no nosso conjunto de dados, um filme, vamos ter onze milhes. E um valor associado de cinco, ponto, alguma coisa milhões. Mas se nós pensarmos, podemos falar que "2x – 7". Será que ela é boa? O que acontece? Nós temos esse menos sete. Nós podemos falar que também é menos cinco, é outra reta. Ou um menos dez, é outra reta. Ou até "5x + 10", é outra reta. Será que ela é boa para mim? Nesse caso, parece que não, porque nossos dados estão nessa reta daqui.

[06:53] Nós precisamos encontrar a melhor reta para podermos fazer isso. Nós temos um processo que é usado em aprendizado de máquina, em machine learning, que é derivado da estatística, que é a regressão linear. E a regressão linear faz isso para nós. Ela encontra a melhor reta para os nossos pontos, e o nome regressão já vem disso. Eu vou querer prever um dado e eu vou cuspir um número em cima disso.

[07:19] Como nós fazemos isso? No primeiro momento, nós precisamos nos preocupar em dividir os nossos dados em dados de treino e dados de teste. Porque não adianta nada nós generalizarmos nosso modelo em cima de todos os dados, porque nós nunca vamos saber quando está bom para um dado que nós nunca vimos antes e nós vamos nos atrapalhar. Ou talvez, quando venha de fato um dado que nunca vimos, nós performamos muito mal.

[07:44] E a ideia é termos um conjunto de dados de teste para validarmos nossa ideia, para ver o quão bom o nosso modelo está. E um conjunto de dados de treino para nós treinarmos em cima desses dados. Como fazemos isso? Em um primeiro momento, nós usamos a biblioteca de machine learning que nós temos no python, que é o scikit-learn. Eu estou abrindo o Atom para justamente nós fazermos isso.

[08:08] Voltei aqui para as minhas explorações. E a ideia é nós a importarmos. Eu tenho “sklearn.model_selection import train_test_split”. Qual é a ideia? Antes de fazermos isso, lembra que eu estava falando que quando olhamos nosso gráfico, nós temos um conjunto de variáveis dependentes, aquele nosso X, que aí nós encontramos aquele M e aquele B. E vamos ter uma variável resposta, uma variável dependente, uma variável que depende disso.

[08:53] Como precisamos fazer? Nós precisamos dividir os nossos dados em dados de treino e dados de teste. Mas antes disso, nós precisamos dividir as nossas variáveis em dados independentes e dados dependentes. Como nós podemos fazer isso? Eu vou ter aqui o meu filmes_investimento. Ele vai receber o meu movies investimento em milhões. Eu vou criar esse dado aqui. Eu estou voltando para o terminal para colar. Colei, voltei para cá novamente.

[09:35] E eu preciso fazer a mesma coisa com o outro cara. Qual é o outro cara? É independente. Filmes, bilheteria. Porque ele recebe movies, bilheteria, pessoas. Copiamos aqui, vamos colar. Se dermos uma olhada, o que temos? Novamente, aqui é o nosso data frame de todos os investimentos, linha a linha. Se fizermos agora o filme, bilheteria, vamos ter a mesma coisa para as bilheterias. Vamos ver se isso é realmente o que acontece?

[10:18] Agora, nós precisamos dividir os nossos dados em dados de treino e dados de teste, como eu falei inicialmente, e como fazemos isso? Importando primeiramente o método. Vamos aqui, ok. Vamos voltar aqui. Salvo. Está importando. Importamos.

[10:50] Agora, precisamos dividir nossos dados em dados de treino e dados de teste. O treino é em cima das nossas variáveis independentes e depois os dados de teste em cima das nossas variáveis dependente. E como é essa divisão? Ela é bem simples.

[11:04] Podemos fazer um “treino, teste, treino_bilheteria, teste_bilheteria”. Esses dados vocês sabem o método treino. Teste Split. O que nós passamos pelo teste Split? Justamente os dados que fizemos. Justamente filmes investimento, que é a nossa variável independente e filmes bilheteria.

[11:43] Agora, vamos pegar esses dados, copiar e dar uma olhada no que eles fazem. O que eles são, na verdade. Copiei os dados, filmes e investimento não está definido. Copiamos aqui. Vamos dar uma olhada? Tem meus dados de treino. Eles são o investimento. Se olharmos as de teste, eles também são investimento. Mas qual é o tamanho? Qual é a diferença desses caras?

[12:09] Se eu olhar o lado de treino, 6843. Se eu olhar o lado de teste, 2282. Se eu somar os dois, 9125. Qual é o tamanho do meu data frame inteiro? 9125. Eles separaram.

[12:35] Quantos por cento é treino? Se eu pegar o tamanho de treino e dividir pelo total, aproximadamente 75%. Setenta e cinco por cento, na verdade. No meu dado de teste, eu tenho 25%. Se eu somar os dois, eu vou ter cem por cento. Vamos ver se é isso que acontece?

[12:57] Vamos digitar de novo? Eu tenho aqui os meus dados de treino. Ele dividido pelos meus dados de teste. Os meus dados de teste dividido pelos meus dados de totais de filmes. Um. Cem por cento. Fizemos a nossa divisão.

[13:20] Agora, o que precisamos fazer? Se dermos uma olhada na fórmula desses caras. Vamos em “treino.shape”. Não tem definido aqui o número de colunas. E novamente aqui é um tipo de series, como já vimos naquele erro. Se eu der uma olhada no treino, ele é um panda, ponto, series.

[13:41] Vamos transformar esses caras em um array, num vetor coluna, para conseguirmos fazer as nossas previsões. Até porque depois fica mais fácil manipular para o machine learn entender.

[13:52] Como fazemos isso? Usamos a biblioteca de manipulação de dados de matrizes, na verdade, do python, que é o numpy. Nós temos aqui o import numpy as np. O que o numpy faz? Ele vai criar um array. Ele vai transformar esses caras num array. É só fazer “treino = np.array(treino)”. E eu estou reformando esse array para aquilo que queremos.

[14:22] O que nós queremos? Nós queremos que seja um vetor exatamente como vimos no Excel. É um vetor coluna. Ele tem seis mil e não se quantas linhas e uma única coluna. Vou dar um reshape, passando o próprio tamanho do número de linhas. Treino. E uma única coluna. Para vermos esse dado de uma forma um pouco mais bonita. Estou copiando ele aqui. Depois, vamos aqui e colamos ele aqui.

[14:55] Np is not defined, não definido, porque nós não importamos a biblioteca. Eu escrevi lá e não importei aqui. Agora, colamos de novo com o command + V. Criamos nossa área de treino. O que acontece agora? Está num formato um pouco melhor para trabalharmos.

[15:10] Vamos repetir esse processo com todos os outros caras. Eu vou ter um treino aqui, tenho o meu "teste = np.array(teste).reshape(len(teste),1)". E "treino_bilheteria = np.array(treino_bilheteria).reshape(len(treino_bilheteria),1)". E "teste_bilheteria = np.array(teste_bilheteria).reshape(len(teste_bilheteria,1)".

[16:10] Vamos copiar todos esses caras aqui. Copiamos e vamos colar. Teste. O erro está em algum dos canais. Teste bilheteria. É bilheteria aqui. Os outros três deram certo. Vou só copiar eles de novo para passar. Legal. Copiamos aqui com o command + C e vamos aqui no command + V. Ok, criamos nossos dados.

[16:41] Se olharmos todos eles, eles vão ser um array. Qual é a ideia, agora que já temos os dados no formato que queremos? Nós vamos criar o nosso modelo, isso é bem simples "LinearRegression()". Criamos o nosso modelo. E nós vamos encaixar os nossos modelos de acordo com os nossos dados de treino.

[17:08] Esse encaixe, para fazer ele aprender de fato, é o "modelo.fit(treino, treino_bilheteria)" Se olharmos aqui, vamos aqui. Modelo, ponto, fit is not defined. Por que não está definido? Porque nós não criamos o modelo como eu digitei ali em cima. O problema de criar o script depois ir pra terminal é justamente esse. "modelo = LinearRegression()". Linear regression not defined também porque eu digitei lá em cima e não importei.

[17:43] Na verdade, eu não importei. Como nós fazemos a importação dos dados desse cara? Nós jogamos um "from sklearn.linear_model import LinearRegression". Agora sim. Nós podemos copiar o cara aqui e nós podemos jogar o modelo criado e aplicar. E olha só que legal. Nós criamos o dado.

[18:16] O que isso fez? Ele basicamente encontrou aquele M e aquele B que eu falei. Novamente, se nós voltarmos aqui para o nosso Geo Gebra, qual a ideia do B? Se nós vermos, eu tenho aqui "2x – 7", por exemplo, que é essa reta. O que acontece se por menos seis? Menos cinco? Menos quatro, menos três? Olha só. A reta sempre se move para cá ou para cá. Nós encontramos esse B onde ele intercepta o eixo Y, o eixo da bilheteria.

[18:53] O resultado que ela dá intercepta, interception, intercept. "modelo.intercept_". Esse é o nosso B, o B que nós encontramos.

[19:04] E o X, o que ele significa? Novamente, eu vou voltar aqui para o caso inicial. E se o meu X fosse 1.9? 1.8? 1.7? 1.6? 1.5? Um, ponto, quatro? 1.3? Ele está mudando o ângulo que essa reta recebe em função comparado com o meu X. Mudando o ângulo, ele é um coeficiente angular. Coeficiente, coef, de coeficiente. "modelo.coef_".

[19:37] A equação da minha reta, nesse caso da reta que nós encontramos, a reta de regressão. A melhor reta nesse caso é vezes um X mais esse valor aqui. Essa é a equação da nossa reta. Vamos ver se é isso que acontece? O modelo, essa variável que acabamos de criar tem um método chamado "modelo.predict()", que nós fazemos a previsão daquilo que queremos.

[20:03] Voltando lá para o nosso dado de Zootopia, que eu tinha aberto aqui do lado. Abrir zootopia data. Nós temos essa aqui já pronta para copiar. Quanto seria a bilheteria? Nós vamos aqui e fez a previsão. Seria aproximadamente 7.8 milhões de pessoas.

[20:24] Vamos ver se batem os valores que nós fizemos? “modelo.coef_*” o valor mais “modelo.intercept_”. “modelo.predict_”. E agora nós queremos o “modelo.coef_”, vezes o valor que queremos mais o “modelo.predict_”. Olha só. É o mesmo valor. Exatamente o mesmo valor. Essa foi a equação da reta que encontramos.

[21:09] Mas nós chegamos para o nosso chefe e falamos: chefe, se nós investirmos vinte e sete milhões, vamos ter aproximadamente 7.8 milhões de pessoas que vão. Será que nós confiamos nesse dado? Será que esse dado é confiável? Será que nós podemos chegar e reportar para o chefe com toda a segurança do mundo?

[21:26] Existem métricas, quando você trabalha com o aprendizado de máquinas, estatística mesmo, existem métricas para termos um grau de confiabilidade do nosso modelo. Uma dessas métricas é o índice de erro. Quanto nós erramos entre os valores que previmos e o valor real.

[21:43] Como nós descobrimos essa métricas? Nós temos os próprios métodos dentro do scikit, mas como nós entendemos o que está acontecendo por trás e se esse valor é confiável ou não, é exatamente o que vamos aprender no próximo vídeo.