7.3 - Introdução a regressão linear (parte 1)

00:00:00:00 - 00:00:01:10
Olá, pessoal. Na aula de hoje, nós vamos treinar o modelo de regressão e vamos avaliar para ver como ficou. Bora lá. Bom, para relembrar, na última aula nós fizemos um trabalho de pré-processamento com os dados que vamos utilizar para o nosso modelo de regressão linear, correto? No final da aula, nós aprendemos que precisamos dividir os dados em teste, treino e também aplicamos a função de normalização. Agora, vamos para a melhor parte, que é treinar o modelo. Nós vamos utilizar o modelo de Linear Regression da biblioteca... Qual biblioteca? Sklearn. E a gente faz o import desse Linear Regression assim: from sklearn ponto linear model import linear regression. Aqui. A gente cria o objeto dessa regressão linear assim: model linear regression. Deixa eu copiar, né? Copiar, colar. Pronto. Criamos o objeto desse modelo e agora vamos treinar. Vocês estão preparados?

00:00:01:10 - 00:00:02:11
A gente vai treinar, tá, gente? Para isso, nós chamamos o modelo e colocamos a função de fit e aí dentro do fit a gente coloca lá os nossos parâmetros. Vamos lá: model ponto fit... E aí dentro dos parênteses a gente coloca os parâmetros. O primeiro são os nossos atributos de treino, que a gente já normalizou aqui bonitinho e o segundo são as respostas, que são os nossos alvos. Então aqui a gente tem aquele rolezinho lá, que a gente está falando os dados, falando os atributos, e está dando o resultado, porque aí o modelo vai treinando. Lembrando que não normalizamos o nosso objetivo-alvo, o nosso target, que é o salário. Porque, veja bem, a gente quer o resultado em valor de salário. Se a gente normaliza, como que a gente vai entender o resultado? Precisaria de uma outra conversão para voltar a ter o valor de salário. Muito complicado. Bom, perfeito. Vamos executar essa célula. Olha só, e temos um modelo treinado.

00:00:02:11 - 00:00:03:46
A gente pode fazer as predições do nosso conjunto de testes utilizando a função do predict, que é assim: o model ponto predict e aí aqui a gente manda um parâmetro só, que é o nosso conjunto de atributos de teste já normalizado. Não teve emoção, né, gente? Vou treinar de novo, tá? Ó, de novo aqui. Agora sim, vocês vão treinar o modelo, clique aqui, modelo treinado. Vamos fazer o predict. Ok. Olha só, quando a gente faz o predict, esse resultado aqui, esses valores são os valores de salário que o modelo predisse para cada conjunto de atributos que ele teve. Vamos pegar esses resultados e colocar num valor, uma variável, na verdade. Vamos chamar essa variável de yPred, ok? Que é y de predição, porque é muito criativo e tudo mais. Após treinar um modelo de Machine Learning, é essencial avaliar o desempenho dele, para entender o quão bem ele está fazendo as predições. Duas métricas comuns para avaliação de modelos de regressão são o erro quadrático médio e o coeficiente de determinação. Vamos lá. O erro quadrático médio, o MSE, É uma métrica que calcula a média de diferença entre o valor predito e o valor real. Entretanto, nessa métrica, a diferença é elevada ao quadrado. Dessa maneira, penaliza valores que sejam muito diferentes entre o valor previsto e o valor real, ok?

00:00:03:46 - 00:00:05:20
Já o R² é uma medida estatística de quão próximos os dados estão da linha de regressão, daquela reta que a gente já viu. A definição do R² é bastante simples, é a porcentagem da variação da variável resposta que é explicado por um modelo linear ou R² é igual a variação explicada dividido pela variação total. O R² está sempre entre 0% e 100%. O 0% indica que o modelo não explica nada da variabilidade dos dados de resposta ao redor da sua média e 100% indica que o modelo explica toda a variabilidade dos dados de resposta ao redor da sua média, ou seja, o foco é o 100%. Surpreendendo zero pessoas, essas funções também são da biblioteca Sklearn. A gente faz o import dessas funções assim: from sklearn ponto metrics, de métricas, import, aí a gente faz o import do mean squared error. Ah, e uma coisa que eu não falei antes eu acho, até porque não teve a oportunidade, mas agora tem a oportunidade, é que a gente vai fazer o import de duas funções de uma mesma biblioteca. A gente vai fazer o import de duas métricas que estão aqui no sklearn ponto metrics. A gente não precisa escrever uma outra linha e repetir tudo. Basta eu colocar uma vírgula aqui e colocar qual que é a outra função que a gente está querendo. No caso, é r2 underline score, ok? E aí vai fazer o import dessas duas funçõezinhas.

00:00:05:20 - 00:00:06:54
Vamos calcular primeiro o erro quadrático médio, o MSE. Esse erro quadrado médio, então, vai calcular a diferença entre os valores reais, que no caso aqui é o nosso Ytest, que são os valores reais, e os valores preditos, que a gente já predisse aqui que é o Ypred. Quanto menor o valor do MSE, mais precisa é a predição do modelo. Então, nesse caso a gente está querendo um valor menorzinho. Então, vamos lá. Vamos chamar de MSE mesmo. Vamos pegar o mean square error, que é a função, colocar os parênteses. Dentro dos parênteses, a gente vai colocar apenas dois parâmetros: os valores de teste, que são os valores reais, e os valores preditos. Então vai ser ytest, ypred. E vamos executar. Ok, vamos ver quanto que ficou esse MSE. O MSE do nosso modelo ficou com um valor absurdamente alto, ou seja, indicando que o modelo não está com predições precisas. Essa métrica é útil quando temos valores de target, alvo, mais incomuns, e que seria interessante que o nosso modelo não errasse. Por outro lado, como os erros são penalizados exponencialmente, os maiores têm um peso maior do que os erros menores. Então, se o nosso modelo faz apenas uma predição muito, mas muito ruim, o MSE irá elevar esse erro ao quadrado e, com isso, esse erro ficará ainda pior e acabaremos achando que o nosso modelo está performando pior do que ele realmente está.

00:00:06:54 - 00:00:08:31
Outro ponto é que a escala do MSE não é a mesma do nosso target, do nosso salário, visto que os erros são elevados ao quadrado. Então, sua interpretação fica mais difícil. Então, ao invés do MSE, a gente pode calcular uma outra métrica, que é a MAE, que é o erro absoluto, o erro médio absoluto, que calcula a média da diferença absoluta. Então, nesse caso, a gente não vai elevar ao quadrado, a gente vai pegar só a diferença entre o valor predito e o valor real, sem esse negócio de levar ao quadrado e tal. Nesse caso, os erros são penalizados linearmente, ou seja, todos terão o mesmo peso na média. A vantagem do MAE em relação ao MSE é que, como a escala é a mesma do target, do salário, do nosso alvo, é mais fácil de ser interpretado. Nada ao quadrado, fica mais tranquilo.

Bom, vamos fazer o import dessa biblioteca também, que também está lá no sklearn ponto metrics, então a gente só coloca vírgula mean absolute error. Executo e ok. Eu vou copiar exatamente como está aqui, só vou mudar: em vez de MSE, vou chamar de MAE. E aqui, em vez de squared vai ser absolute e os parâmetros são esses também: são os valores verdadeiros, os valores reais e os valores preditos. Vamos executar e vamos dar uma olhada no MAE. Antes de a gente discutir esse valor de MAE, já vamos aproveitar e dar uma olhada no R^2. Então, vamos lá.

00:00:08:31 - 00:00:09:33
R^2 é igual ao R2 score, e também são os mesmos parâmetros. Todas as funções de métricas recebem os mesmos parâmetros: os valores reais e os valores preditos. Vou colocar aqui R2. Ok. Então, para saber se o nosso modelo está bom, precisamos analisar as métricas em conjunto. Temos um R2 de 0.52, um MAE de 3291 e um MSE de 23 milhões. Então temos uma variação razoável. Na média absoluta, os erros não são tão altos, pois 3 mil para as faixas salariais que temos é até aceitável, porém temos um MSE bem alto, o que significa que temos bastantes erros, que sim são altos. Para um modelo real de produção, nosso modelo não seria aceitável, mas vamos olhar um pouco melhor os resultados para entender se conseguimos usar ele para realizar algumas análises.