7.2 - Preparação dos dados (parte 3)

00:00:00:00 - 00:00:01:37
Bom, e é isso, a gente fez as alterações que foram necessárias nas colunas individuais, agora a gente pode fazer a seleção das colunas que a gente vai utilizar para o modelo mesmo, que vão ser os atributos do nosso modelo, tá? Então vai ser: dados, aí eu vou colocar dois colchetes, deixa eu fazer uma coluna aqui em cima para poder pegar o nome das colunas todas, que eu não lembro, e vamos lá. A gente vai ter idade. A gente vai ter gênero. Bom, idade e gênero a gente já tinha combinado que já seria. Depois, ao invés de ser cor, raça e etnia, vai ser aquela coluna que a gente criou, que é a coluna não branca. Então aqui: vírgula não branca. Uma outra que a gente também já criou foi a tempo de experiência. Então, vai ser essa coluna tempo de experiência aqui. A outra coluna pode ser essa de insatisfação aqui também que a gente criou, né? Está no finalzinho. Depois vai ser a coluna de setor. Setor é importante também, né? Saber qual setor que a pessoa trabalha. Às vezes o setor influencia no salário dessa pessoa. A região também é um outro fator que pode influenciar. Às vezes, uma pessoa de uma região X ganha mais que uma de uma outra região. Então região onde mora. Vou colocar aqui.

00:00:01:37 - 00:00:03:12
O nível de ensino que a gente acabou de trabalhar aqui. Deixa eu dar um enter aqui e continuar aqui embaixo. O número de funcionários também, que a gente tinha mexido. Aqui: número de funcionários. E por último o nosso salário, né? Aqui. Ah, tem o nível de senioridade também, né, gente? Aqui a gente criou uma coluna em uma aula que agora não é nível, é novo nível. Então vamos pegar aqui, ó. Coluna novo nível. Então ok, a gente vai ter todas essas colunas como atributo. Se eu der um Enter aqui, vocês vão ver que tem só essas colunas. Então vou atribuir ao dados apenas essas colunas, ok? Bom, agora se a gente der um dados ponto columns, a gente vai ver que temos só essas colunas. Agora a gente vai utilizar o get dummies para as colunas de gênero, região onde mora, novo nível e setor. Nesse caso, usaremos realmente o get dummies pois essas variáveis não têm uma ordenação. Por exemplo, eu não posso falar que masculino vale mais que feminino. E como o modelo vai criar multiplicadores para cada variável, não faria sentido ter valores numéricos aqui. A gente poderia substituir por valores, mas como estamos usando também uma análise das importâncias, acabaria influenciando na nossa análise final.

00:00:03:12 - 00:00:04:46
Bom, então vamos usar o get dummies. Como é que é o get dummies mesmo? Ele é do Pamdas, né? Então, pd ponto get dummies aqui. E aí dentro dos parênteses, o primeiro parâmetro é a nossa tabela, que é dados. Depois a gente vai colocar as colunas que a gente vai fazer o get dummies. No caso, columns igual colchetes e dentro vai ter as colunas. Gênero, depois região onde mora, novo nível e setor, né, gente? Então setor, novo nível e região onde mora. Deixa eu copiar isso. Todas eu escrevi, digitei uma por uma. Na última, copiei. E um último parâmetro aqui, vamos apagar a coluna inicial para não ficar aquele tanto de coluna a mais. Então, drop first, que é dropar, cortar a primeira coluna, true. Vamos só rodar aqui. Vamos ver. Bom, ó. Bonitinho. Se a gente arrastar aqui para o final, a gente vai ver as colunas que foram criadas, né? Bacana. Vamos atribuir esse resultado a dados.

Perfeito, temos a tabela com os dados, mas antes de criar o modelo em si, precisamos separar o nosso conjunto de dados em dois. Uma parte vai ser o conjunto de treinamento, que é utilizado pelo modelo para aprender mesmo, e a outra parte é o conjunto de teste, utilizado para avaliar o desempenho do modelo para os dados que o modelo ainda não viu.

00:00:04:46 - 00:00:06:24
Então, primeiro vamos separar os atributos do nosso objetivo. O atributo: setor, gênero e tal. O objetivo, que é o salário. Então, vamos lá. Colocar x, que são os nossos atributos, dados ponto drop, que é para excluir, salário. Então, a gente está excluindo o salário. A gente vai colocar o axis igual a um só para indicar que é a coluna que a gente está apagando. Então, ok. E o Y aqui vai ser dados e a coluna salário. O que a gente fez aqui? A gente dividiu em atributos e target, que é o nosso alvo. Peguei todos os atributos e tirei a coluna de salário, e o nosso target eu peguei só a coluna de salário, tá? Bom. Colocamos a variável Y apenas na coluna salário, a variável X todas as outras variáveis da tabela. Para fazer a separação do conjunto de teste e treino, vamos utilizar uma função do Sklearn, também chamada de train test split. Basicamente é uma divisão de treino e teste. A gente faz o import assim... Deixa eu rodar essa daqui... From sklearn ponto model selection import train test split.

00:00:06:24 - 00:00:08:02
Então lá na biblioteca do sklearn da parte de seleção de modelo, model selection, eu estou importando a função de train test split. A gente chama a função bem direta: train test split. Dentro dos parênteses, a gente envia os dois primeiros parâmetros, que primeiro são os atributos e depois o target, o nosso alvo. Então, primeiro são os atributos X e depois o Y. O próximo atributo se chama test size, que é basicamente o tamanho do meu conjunto de teste. Esse parâmetro recebe o valor em porcentagem. Então, por exemplo, vamos dizer que o nosso conjunto de teste é 20% do total do meu conjunto. Então, a gente coloca assim: teste size igual a 0.3. Então, isso significa que 20%, 0.2. O 3 apareceu ali e me influenciou, vocês viram? Então, isso significa que 20% dos dados serão usados para o conjunto de teste e 80% para o conjunto de treinamento. O último parâmetro que a gente vai utilizar é o random state. Random state. Random state é uma semente para o gerador de números aleatórios, que garante que a divisão dos dados seja reproduzível. Porque, pensem comigo, essa divisão é realizada de forma aleatória. E é como se a gente embaralhasse todas as linhas e colocasse 20% para teste e 80% para treino.

00:00:08:02 - 00:00:09:30
Aí beleza, treinamos o modelo e esquecemos de salvá-lo. Como que a gente vai fazer para gerar exatamente o mesmo modelo, sendo que essa divisão dos dados é aleatória? Por isso a gente usa esse Random state, para garantir que essa exata divisão seja reproduzível. A gente pode colocar qualquer número nesse parâmetro. Só lembrando que se quiser reproduzir essa mesma divisão, sem embaralhar de novo os dados, você usa o mesmo número. Vou colocar aqui 42, e se vocês olharem muito, a gente usa 42, porque dizem que é a resposta para a questão fundamental da vida, do universo e tudo mais. Nossa, piadinha nerd no meio da aula, né? Nossa, que situação, a que ponto cheguei! Ótimo, o resultado dessa função vai ser dois conjuntos de atributos, um para teste e um para treinamento. E dois conjuntos com o alvo, target, um para teste e um para treinamento.

Então a gente coloca assim, ó: vou colocar o igual aqui, vou pegar o mouse para ajudar. Então, atributos primeiro: xtrain, xtest. Ok. E agora dois conjuntos para o target, o nosso alvo: ytrain, ytest. Ok. Train é de treino e test é de teste, tá? Para finalizar essa aula, vamos utilizar uma função chamada Standard Scaler que, adivinhem, também é da biblioteca Sklearn. Deixa eu rodar essa linha aqui, né?

00:00:09:30 - 00:00:11:15
Fico esquecendo. Rodei. A gente faz o import dessa Standard Scaler assim: from sklearn ponto preprocessing import standard... Opa. Como é que escreve? Standard scaler. Ah, apareceu aqui para mim. Scaler. Ok. Essa função tem como objetivo padronizar as características, removendo a média e escalando para a variança unitária. Traduzindo, ela vai normalizar os nossos dados. É importante fazer isso pois algoritmos de machine learning, como regressão linear e redes neurais funcionam melhor quando os dados têm uma escala uniforme. Então a gente cria um objeto de normalização, scaler, por exemplo, a gente coloca standard scaler. A gente criou esse objeto e aí a gente aplica. Como? A gente pega ele: scaler ponto fit e aí a gente coloca como parâmetro os atributos: xtrain. Na verdade, não é fit apenas, é fit transform. Tem um underline aqui. Fit transform. Esse resultado a gente joga numa variável. Vou chamar de xtrain, mas para não confundir com outro, não sobre-escrever, vou chamar de xtrain scaled. É como se fosse os atributos da parte de treino normalizados, ok? E vou fazer a mesma coisa com os atributos de teste.

00:00:11:15 - 00:00:12:01
Eu copiei e colei aqui e vou só substituir para test onde tem train. Ok, vou executar. E pronto. Não precisamos necessariamente apontar quais as colunas que queremos normalizar porque a função normaliza o que precisa ser normalizado. Beleza? E é isso tudo, gente. Nessa aula a gente fez um trabalho de Featuring Engineering que começou na última aula, continuou nessa. Para deixar os nossos atributos no formato certinho para o modelo, a gente usou técnicas que a gente já viu e algumas técnicas que são novas. Além disso, a gente aprendeu um pouco mais sobre pré-processamento de dados para modelos de regressão. Na próxima aula, nós vamos treinar o modelo e avaliar como que ficou. Até lá!