Regressão Logística

Olá, seja bem-vindo!

Você já deve estar familiarizado com a ideia de que os algoritmos de classificação são utilizados para solucionar problemas que envolvem tomada de decisão, e que as variáveis de saída, nesse caso, são categóricas, ou seja, a resposta do algoritmo pode ser representada como 0 ou 1, verdadeiro ou falso, ou definida em outro tipo de label, dentre uma lista de labels possíveis.

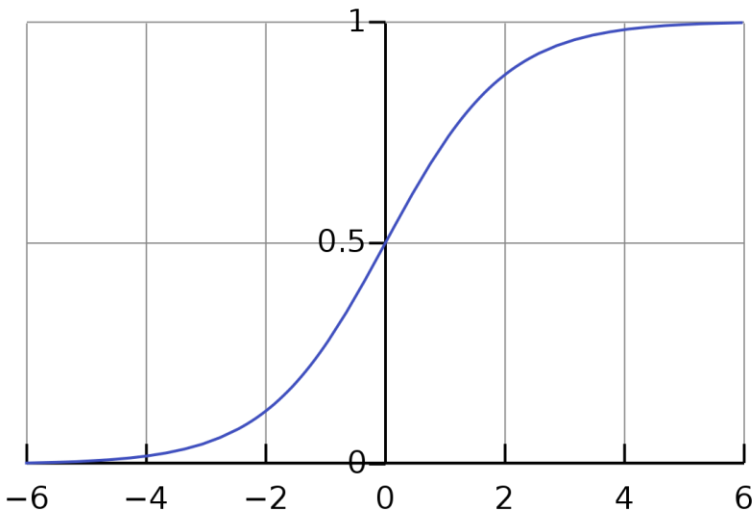
Então, chegou o momento de você conhecer um dos algoritmos mais usados em problemas de classificação: a regressão logística. Ela possui vários cenários de aplicações, podendo ser utilizada, por exemplo, na área da saúde, caso deseje criar um sistema que classifique remédios que podem ser indicados para pacientes com determinadas enfermidades; ou por instituições financeiras, na predição de risco de inadimplência ou análise e liberação de crédito. Em outras palavras, é enorme o leque de possibilidades, não acha?

Portanto, a regressão logística busca estimar a probabilidade de a variável dependente assumir uma determinada classe ou categoria. Diferente da regressão linear, cuja resposta é um valor contínuo, a regressão logística apresenta uma resposta categórica, em que o modelo escolhe alguma das categorias que já existem nos dados trabalhados.

Matematicamente, a função utilizada nesse tipo de regressão é a logística, que também é conhecida como sigmóide, pois, graficamente, possui um formato de “S” e é representada pela equação: $y = 1/(1 + \exp(-x))$. Essa equação não é sensível a valores extremos, e é, justamente, essa a vantagem desse método, se tornando, assim, muito usual para lidar com outliers, valores existentes em um conjunto de dados que são muito discrepantes dos demais.

Então, a função logística retorna valores entre 0 e 1, ou seja, valores em termos de probabilidade. Desse modo, o modelo de regressão utiliza essa função para descobrir a probabilidade de um valor x estar em determinada categoria.

Em uma classificação binária, por exemplo, em que só há duas categorias, 0 e 1, o modelo de regressão logística classifica os valores retornados pela função abaixo de 0.5 como sendo na categoria 0; já os valores iguais e acima de 0.5, como sendo da categoria 1. Assim, a resposta final desse modelo é uma das duas categorias. Como é demonstrado no gráfico que apresenta uma curva em formato de S, e é crescente a partir do ponto (-6, 0) que, representando os valores dos eixos x e y, respectivamente, cruza o eixo y no valor 0.5 e continua crescente até se tornar constante a partir do ponto (6, 1).



Para facilitar ainda mais o seu aprendizado, acompanhe o exemplo prático, a seguir, no ambiente do jupyter. O objetivo é construir um modelo de machine learning para predizer se uma paciente tem ou não diabetes. Para isso, será usado os dados coletados pelo Instituto Nacional de Diabetes e Doenças Digestivas dos Estado Unidos. Dados, baseados nas informações sobre as pacientes adultas, como número de gestações, pressão sanguínea, idade etc.

Inicialmente, com o ambiente do jupyter aberto, você deve importar as bibliotecas numpy, pandas e matplotlib. Para isso, deve escrever, na primeira linha, o comando “import numpy as np”; na segunda, “import pandas as pd”; e, na terceira, “import matplotlib.pyplot as plt”.

Após importar os módulos, realize a leitura dos dados com informações das pacientes que possuem ou não diabetes. Então, crie um Dataframe do pandas e use o método pd.read_csv para ler o arquivo “diabetes.csv”, que já deve constar em seu computador. Logo, será possível escrever “df = pd.read_csv('diabetes.csv')” e, para verificar as primeiras linhas do arquivo, o comando df.head()

Após executar as linhas de códigos, serão exibidos, em tela, os dados organizados em uma tabela de 9 colunas e quatro linhas, exibindo valores. A primeira coluna, **Pregnancies,** indica a quantidade de vezes que uma paciente ficou grávida; a segunda, **Glucose**, informa a concentração de glucose; **BloodPressure**, **SkinThickness**, **Insulin** e **BMI** correspondem, respectivamente, à pressão sanguínea, à espessura da pele, ao nível de insulina e ao índice de massa corpórea. A sétima coluna, **DiabetesPedigreeFunction**, representa uma pontuação da probabilidade de diabetes com base no histórico familiar. Já a coluna **Age** contém a idade das pacientes; e, por fim, a nona coluna **Outcome** é a variável que deseja-se prever e que representa se a pessoa teve diabete com o valor 1 e se não teve com valor 0.

Bom, a partir daqui, você pode extrair algumas informações estatísticas dessa base de dados. Para isso, escreva o comando “df.describe()” que retorna uma tabela com diversas informações, como, por exemplo, é possível verificar que 75 % dessas pacientes tiveram diabetes, e que 75 % estão na faixa de 41 anos de idade.

Também é possível gerar um gráfico com as idades das pacientes versus o número de vezes que elas ficaram grávidas. Para isso, use a biblioteca do matplotlib, escreva o comando “plt.plot(df['Age'], df['Pregnancies'], 'o')”. Para adicionar um rótulo no gráfico, referindo-se ao eixo x, escreva “plt.xlabel(‘Idade’)” e, para plotar no gráfico o título do eixo y, insira “plt.ylabel(‘Gravidez’)”. Assim, as descrições Idade e gravidez irão identificar os eixos x e y, respectivamente. Para finalizar e exibir o gráfico, escreva plt.show(). Pronto! Você terá, em tela, um gráfico de dispersão com a relação entre idade e gravidez, a partir dos dados importados disponíveis para uma análise.

Após essa análise exploratória, você deve estar pronto para separar os dados em dados de treino e teste, uma parte deles para treinar o algoritmo e a outra para testá-lo com novas informações.

Em primeiro lugar, separe da base de dados a variável dependente y, aquela que deseja prever. Para isso, escreva “y = df['Outcome']”, cuja resposta é 0, se a pessoa não tem diabete; e 1, se tem. Feito isso, remova a coluna **'Outcome'** da base de dados, que será representada pela variável x, escrevendo “x = df.drop('Outcome', axis=1)”. Logo, x conterá todas as colunas do Dataframe exceto a coluna **'Outcome'**. Escreva “x.head()” para verificar se a coluna **'Outcome'** realmente não está na variável x.

Agora, para separar os dados em treino e teste use o método “train_test_split” da biblioteca sklearn. Então, escreva a linha de código “from sklearn.model_selection import train_test_split” para importar o módulo. Já para fazer a separação dos dados escreva “x_treino, x_teste, y_treino, y_teste = train_test_split(x, y, test_size=0.2)”. Assim, 80% dos dados serão usados para treinar o algoritmo e 20% para testá-lo. Sendo assim, o parâmetro “test_size” da classe “train_test_split” define a porcentagem de dados usados para teste.

Para realizar as previsões, será usado o modelo “LogisticRegression” da biblioteca sklearn. Então, para importar o modelo, escreva “from sklearn.linear_model import LogisticRegression”. Agora, é só criar o modelo, escrevendo a linha de código “modelo = LogisticRegression(max_iter=5000)”, em que a variável modelo é o modelo de regressão logística com o número máximo de 5000 iterações.

Então, para realizar esse treinamento, utilize o método fit da classe modelo. Escreva “modelo.fit(x_treino, y_treino)”, passando para ele as variáveis x_treino e y_treino.

Treinado o modelo, agora, você pode fazer as previsões, usando o método predict da classe modelo. Para isso, escreva “y_previsto = modelo.predict(x_teste)”, passando a variável x_teste para fazer as previsões. Desse modo, a variável y_previsto contém o resultado das previsões de diabetes para as pacientes. Então, para mostrar os 10 primeiros valores previstos, escreva “print(y_previsto[:10])”. Analisando esses resultados, é possível identificar se existe uma nova paciente que pode ser classificada com diabetes.

Para saber o quanto esse modelo é capaz de fazer as previsões corretamente, você pode usar o método “accuracy_score” da classe sklearn.metrics. Agora, para importar o método, escreva “from sklearn.metrics import accuracy_score” e, para calcular a acurácia do modelo, escreva “accuracy = accuracy_score(y_teste, y_previsto)”, em que as variáveis y_teste e y_previsto são utilizadas para calcular o score.

Assim, como resultado, tem-se que esse modelo é capaz de prever cerca de 80 % dos resultados corretamente. Em se tratando de uma doença grave, esse modelo não é adequado para sobrepor qualquer avaliação médica, mas pode ser usado para auxílio em pesquisas.

Interessante, não é mesmo?

Até aqui, você compreendeu como funciona a técnica de regressão logística, e como ela pode ser representada, matematicamente, além de abordar as suas amplas aplicações. Por último, entendeu como criar um modelo de predição para prever se uma pessoa tem ou não diabetes com base em um conjunto dados clínicos. Ah, não esqueça de revisar todo o conteúdo, aprofundar os seus conhecimentos e resolver os exercícios propostos.

Bons estudos e até mais!