O que é uma regressão logística?

A regressão logística é um modelo estatístico usado para estimar a probabilidade de ocorrência de um evento binário. Em termos simples, é aplicada quando desejamos prever se um evento acontecerá ou não, baseado em uma série de variáveis, como, por exemplo, prever se uma pessoa sobreviverá ou não, baseado no sexo, classe, etc.

A essência da regressão logística consiste em converter uma série de valores (uma linha em um dataframe por exemplo) em uma forma binária, 0 ou 1. No nosso contexto, ela analisa as informações relevantes sobre um passageiro, como classe, título e sexo, atribuindo pesos a esses dados e, por meio desse processo, resumindo-os em uma previsão binária de 0 ou 1.

Por que usar Regressão Logística e não Regressão Linear?

A regressão linear busca compreender a relação entre duas ou mais variáveis. Ao receber uma série de valores X, como classe, título e sexo, ela tenta prever uma variável Y, como "Sobreviveu". Contudo, a limitação é que a regressão linear não consegue classificar explicitamente entre "morreu" ou "sobreviveu"; ela produz um valor que pode não ter uma interpretação direta.

A regressão logística, por sua vez, opera de maneira semelhante à regressão linear, mas com uma distinção, ela ‘força’ o valor de Y para ficar entre 0 e 1. Além disso, a regressão logística utiliza um sistemas de pesos baseados na probabilidade de um evento ocorrer, enquanto a regressão linear utiliza a correlação entre as variáveis.

As matemáticas (os exemplos são baseado no titanic):

A fórmula da regressão logística é dada por:

$$\frac{e^{(b0 + b1*x1 + b2*x2 \ldots)}}{1 + e^{(b0 + b1*x1 + b2*x2 \ldots)}}$$

onde basicamente:

os betas (b0, b1, … , bn) serão os nossos pesos baseados em probabilidades, e os x, serão se a nossa variável está presente ou não.

Por exemplo:
No caso do sexo, temos que se a pessoa for uma mulher então x = 1, e se for homem x = 0. Isso basicamente nos diz que se a pessoa for mulher nós consideramos o peso que atribuímos ao sexo.

Então como o nosso objetivo é transformar uma série de variáveis (uma linha do dataframe) em um valor só, nos somamos todos os betas multiplicados pela presença da nossa variável:

Considerando apenas sexo e classe, e como classe temos 3 opções: 1 classe, 2 classe, 3 classe, temos que criar 2 variaveis para classe, x2 (se for 1 classe = 1, se for 3 = 0) e x3 (se for 2 classe = 1 se não = 0)

Por fim temos b0 que é a base de tudo, ele é um indicador de quando tudo for 0, depois eu explico melhor

Então temos:

x = b0 + b1 * x1(sexo) + b2 * x2(2 classe) + b3 * x3(1 classe)

Substituindo com os pesos reais (depois eu explico o cálculo)

x = -0.48 + 2.51 * x1 + 1.03 * x2 + 1.66 * x3

vamos supor o exemplo de uma mulher que está na primeira classe:

x = -0.48 + 2.51 * 1 + 1.03 * 0 + 1.66 *1

x= 3.69

agora que transformamos a nossa linha em apenas um número, é só substituir na fórmula:

$$resultado = \frac{e^{(3.69)}}{1 + e^{(3.69)}}$$

resultado = 0.9756

pensando o mesmo para um homem da terceira classe temos que:

resultado = 0.3822

Agora, temos que pensar um limite. Ou seja, um valor que irá definir se o resultado encontrado se transformará em 1 ou 0, no caso, eu tinha escolhido um valor de 0.9 (porque eu quis)

então teremos que:

mulher da primeira classe = 1 (sobreviveu)

homem da terceira classe = 0 (morreu)

Probabilidade

Como dito anteriormente, a regressão logística utiliza a chance de algo acontecer para calcular os pesos, por isso primeiramente calculamos a probabilidade de algo acontecer.

a probabilidade é dada por:

$$p = \frac{eventos\ que\ aconteceram}{total\ de\ eventos}$$

no nosso caso:

$$p = total\ de\ sobreviventes$$

levando em consideração o sexo feminino:
sobreviveram = 233, total de mulheres = 314

$$p = 233 \,/\, 314$$

$$p = 0.742$$

Chance (odds):

Agora que temos a probabilidade podemos calcular a chance de um evento acontecer, que é dada por:

$$odds = \frac{P(evento)}{1-P(evento)}$$

no nosso caso:

$$odds = \frac{0.742}{1-0.742}$$
$$odds = 2.8765$$

Mas qual a diferença entre a chance e a probabilidade? Bom, basicamente a maior diferença é o jeito que nós expressamos elas numericamente. Ou seja, a probabilidade é um número entre 0 e 1, enquanto a chance é um número qualquer, mas basicamente tem o mesmo objetivo.

Olhando para o código a função odds já calcula a probabilidade e a chance:

```
def odds(total, acerto):
    probabilidade = acerto / total
    odd = probabilidade/(1-probabilidade)
    return odd
```

Logg-odds (logit):

Agora que temos a nossa chance temos que transformar ela em likelihood (não tem uma tradução boa).

Mas por que devemos fazer isso?

https://medium.com/swlh/probability-vs-likelihood-cdac534bf523

Para transformar é bem fácil (lembrando que esse log é na base natural):

$$logit = log(chances)$$

Cálculo dos pesos:

Para calcular os pesos, é necessário escolher a variável que vamos trabalhar, pensando ainda na variavel sexo, devemos pensar primeiro em quem tem a maior chance de sobrivencia, dividir essa chance pela outra, e por fim calcular o logit disso:

Resumidamente:

$$chance(sexo) = \frac{chance(mulher\ sobreviver)}{chance(homem\ sobreviver)}$$

por fim:

$$b(sexo) = log(chance(sexo))$$

no código quem faz esse cálculo é o:

```
def calculaBetai(odds1, odds2):
    divisao = odds1 / odds2
    return np.log(divisao)
```

E para o b0 (intercept)?

É mais simples ainda, é só calcular o loggit da chance de todo mundo sobreviver

$b0 = log(chance(sobreviver))$

no código quem faz esse cálculo é o:

```
#Intercept X = 0
def calculaIntercept(total, acerto):
    return np.log(odds(total, acerto))
```

A fórmula final:

Com todos os pesos em mão, e todas as variáveis, é só inputarmos esses valores na fórmula é calcular a regressão logística para uma linha

X = $b0 + b1 * x1 + b2 * x2$ ...

$$\frac{e^{(X)}}{1 + e^{(X)}}$$

O código:

Por fim quem junta tudo isso no código é a função logística que calcula o valor de uma linha:

```
#Calcula e retorna X
def logistica(series,b0, b1, b2, b3, b4, b5, b6, b7):
    X = 0
    X += b0
    #Sexo
    if(series.iloc[0] == 1):
        X += b1
    #2 Classe
    if(series.iloc[2]== 2):
        X += b2
    #1 Classe
    if(series.iloc[2] == 1):
        X += b3
    #Miss
    if(series.iloc[3] == 1):
        X += b4
```

E quem calcula a regressão logística é:

```
def regressaoLogistica(series, b0, b1, b2, b3, b4, b5, b6, b7):
    exp = np.exp(logistica(series, b0, b1, b2, b3, b4, b5, b6, b7))
    return (exp/(1 + exp))
```

Por essa função roda o dataframe inteiro (podendo ser dividido aleatoriamente entre teste e treino), e guarda os valores achados em um vetor, os valores reais em outros. Assim como dado um limite já transforma os valores achados em 0 ou 1. Por fim, ela retorna um dataframe contendo todos os valores para comparação:

```
def returnDf(limite, dfTreino, dfTeste):
    b0 = returnB0(dfTreino)
    b1 = returnB1(dfTreino)
    b2, b3 = returnB2B3(dfTreino)
    b4, b5, b6, b7 = returnB47(dfTreino)

    data = []
    survived = []

    for i in range(len(dfTeste)):
        data.append(regressaoLogistica(dfTeste.iloc[i],b0, b1, b2, b3, b4, b5, b6,
         b7))
        survived.append(dfTeste['Survived'].iloc[i])
```

links auxiliares:
https://www.kdnuggets.com/2018/02/logistic-regression-concise-technical-overview.html

https://www.youtube.com/watch?v=yIYKR4sgzI8&t=28s&pp=ygUTbG9naXN0aWMgcmVncmVzc2lvbg%3D%3D

https://www.youtube.com/watch?v=YMJtsYIp4kg&t=191s&pp=ygUxIExvZ2IzdGljIFJlZ3Jlc3Npb24gLSBUSEUgTUFUSCBZT1UgU0hPVUxEIEtOT1chIA%3D%3D (dedução da regressão)