

# Estruturando o projeto

A ideia é conseguir um score competitivo na competição Titanic do Kaggle, o intuito é o de ML, mas sem usar o pacote sklearn, vamos utilizar apenas matemática, álgebra e estatística, criando os modelos todos da unha.

A estrutura, a princípio é:  
**EEDA básica**, para entendermos algumas distribuições e o quanto impactam nas distribuições da target. Decomposição de variáveis, usando PCR, que nada mais é que, PCA + regressão, a ideia é reduzir dimensionalidade do dataset e consequentemente, do modelo.

Cross-validation, basico, etc  
onde realmente for necessário, como  
encontrar k-componentes da PCA e  
split do dataset de treino / validação  
Implementar o modelo logístico  
e da regressão do zero  
Treinar, Testar, validar, submit  
e rezar.

Todas as ideias que vou usar  
foram apresentadas até o capítulo 7  
do livro An Introduction to  
Statistical Learning.

Na EDA, vimos diferenças  
claras de escala (exige padronização)  
e proporções, dado que aproximada-  
mente 60% das pessoas não sobrevive-  
ram, exigindo um método mais robusto  
de amostragem para split do que  
uma simples Amostragem Simples Alea-  
tória (AAS), optei por uma  
Amostragem Estratificada com alocação  
proporcional. Vimos também que o proble-

proporcionar dados, com isso um problema de dados faltantes, em Embarked e Age, p/ Embarked optei por preencher os dados faltantes pela moda, já p/ Age, Enriqueci o projeto preenchendo os NA com regressão.

As amostragens N° de filhos e N° de famílias serão combinadas em duas outras: Tamanho da família (Numérico) e Esta sozinho (binário).

Tamanho da família = N° filhos + N° familiares

Esta sozinho } 1, se Tamanho da família = 0, C.C

Os dados mostram um poder de discrepância melhor para Esta sozinho.

Manipulando

Apois preencher os dados faltantes de Embarked pela moda, chegou a vez de Age. Para isso, primeiro

precisamos separar os dados a fim de evitar data leakage, vamos dividir em dois estratos: sobreviven ou não, vamos dividir 70/30, ou seja, 70% dados de treino e 30% de teste p/ o cálculo dos estratos:

$$\text{Treino: } h_h = 0,7 \cdot \frac{N}{N_h} = 0,7 N_h$$

onde  $N$  = número de obs.  $N_h$  = nº obs favoráveis. D/ cada estrato, faremos uma LAS e depois juntaremos esses dados, p/ os dados de teste, pegaremos o restante que não foi escolhido p/ treino.

Depois do split, calcularemos uma Regressão Múltipla usando apenas os dados numéricos, para evitar complexidade de variáveis dummy e etc.

O problema da regressão é estimar um  $\hat{y}$  que seja combinação linear de  $X$  e que a diferença

de  $\hat{Y}$  e  $Y$  seja mínima, ou seja,  
 $\hat{Y} = \beta_0 + \beta_1 X$ , e  
 $e = Y - \hat{Y}$

procuramos estimar  $\beta_0$  e  $\beta_1$ , usando  
o método de Mínimos Quadrados:

$$e_i^2 = (Y_i - \hat{Y}_i)^2 \quad e,$$

$rss = e_1^2 + e_2^2 + \dots + e_n^2$ , queremos  
encontrar  $\min(rss)$ :  $rss = \sum_{i=1}^n e_i^2$   
 $= \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$

$$= \sum_{i=1}^n (Y_i - (\beta_0 + \beta_1 X_i))^2$$

$s(\beta_0, \beta_1) = \sum_{i=1}^n (Y_i - \beta_0 - \beta_1 X_i)^2$ , justamente  
o que queremos minimizar  
Poisso, vamos derivar e igualar a zero  
em respeito a  $\beta_0$  e  $\beta_1$ :

$$\frac{\partial s}{\partial \beta_1} = \sum_{i=1}^n 2(Y_i - \beta_0 - \beta_1 X_i)(-1)$$

$$\frac{\partial s}{\partial \beta_0} = 2 \sum_{i=1}^n -Y_i + \beta_0 + \beta_1 X_i, \text{ igualando a } 0$$

$$\Rightarrow \sum_{i=1}^n -Y_i + \sum_{i=1}^n \beta_0 + \sum_{i=1}^n \beta_1 X_i = 0$$

$$\Rightarrow h\beta_0 - \sum_{i=1}^n y_i + \beta_1 \sum_{i=1}^n x_i = 0$$

$$\Rightarrow h - h\beta_0 = \sum_{i=1}^n y_i + \beta_1 \sum_{i=1}^n x_i$$

$$\Rightarrow \beta_0 = \frac{1}{n} \sum_{i=1}^n y_i - \frac{\beta_1}{n} \sum_{i=1}^n x_i$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

$$\frac{\partial S}{\partial \beta_1} = \sum_{i=1}^n 2(y_i - \beta_0 - \beta_1 x_i)(-x_i)$$

$$= -2 \sum_{i=1}^n x_i y_i - \beta_0 x_i - \beta_1 x_i^2$$

$$= 2 \sum_{i=1}^n \beta_1 x_i^2 + (\beta_0) x_i - x_i y_i \rightarrow \bar{y} - \beta_1 \bar{x}$$

$$\Rightarrow \sum_{i=1}^n \beta_1 x_i^2 + (\bar{y} - \beta_1 \bar{x}) x_i - x_i y_i = 0$$

$$\Rightarrow \sum_{i=1}^n x_i (\beta_1 x_i + \bar{y} - \beta_1 \bar{x} - y_i) = 0$$

$$\Rightarrow \sum_{i=1}^n x_i (\beta_1 (x_i - \bar{x}) + \bar{y} - y_i) = 0$$

$$\Rightarrow \sum_{i=1}^n x_i \beta_1 (x_i - \bar{x}) + x_i (\bar{y} - y_i) = 0$$

$$\Rightarrow \beta_0 + \sum_{i=1}^n x_i(x_i - \bar{x}) + \sum_{i=1}^n x_i(y_i - \bar{y}_i) = 0$$

$$\Rightarrow \beta_0 \sum_{i=1}^n x_i(x_i - \bar{x}) = \sum_{i=1}^n x_i(\bar{y} - \bar{y}_i)$$

$$\Rightarrow \beta_0 = \frac{\sum_{i=1}^n x_i(\bar{y} - \bar{y}_i)}{\sum_{i=1}^n x_i(x_i - \bar{x})}$$

$$\Rightarrow \beta_0 = \frac{\sum_{i=1}^n \bar{y} - \bar{y}_i}{\sum_{i=1}^n \bar{x} - x_i}$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

Esse  $\beta_1$  ta errado, o certo é:

$$\beta_1 = \frac{\sum_{i=1}^n (y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

Mas fai quase

Porém, isso não nos cabe agora, pois precisamos de 5 coeficientes, para isso, vamos usar conceitos de álgebra:

Traduzindo nosso problema:

$$y = \beta_0 + \beta_1 x + \epsilon$$

$\|Y - \hat{Y}\|^2$ , queremos min  $\|Y - \hat{Y}\|^2$ , em outras palavras, queremos a projeção de  $\hat{Y}$  no espaço coluna de  $X$  que minimize a distância do vetor de estimativas com o vetor de valores observados:

$$\|Y - \beta X\|^2 = (Y - \beta X)^T (Y - \beta X)$$

$$= Y^T Y - Y^T \beta X - \beta^T X^T Y + \beta^T X^T \beta X$$

$$= Y^T Y + \beta^T X^T \beta X - 2Y^T \beta^T X^T, \text{ derivando}$$

$$\nabla_{\beta} f(\beta) = (2X^T X)\beta - 2X^T Y, \text{ igualando a zero}$$

$$2X^T X \beta = 2X^T Y$$

$$X^T Y = (X^T X)\beta$$

$$(X^T X)\beta = X^T Y$$

$$(X^T X)^{-1}(X^T Y) = (X^T X)^{-1}(X^T Y)$$

$$\hat{\beta} = (X^T X)^{-1}(X^T Y)$$

$$\hat{\beta} = (X^T X)^{-1} X^T Y, \text{ agora}$$

$\hat{Y} = X\hat{\beta} \rightarrow \hat{Y} = X(X^T X)^{-1} X^T Y$ , dizemos que a matriz mudança de base é dada por  $P = X(X^T X)^{-1} X^T$  e então

$$\hat{Y} = PY$$

Agora, para usar a regressão múltipla, precisamos:

Padronizar os dados

Adicionar a coluna de intercepto na matriz  $X$ .

Para padronizar, usamos:

$\hat{z} = \frac{x_i - \bar{x}}{\sigma}$ , onde  $\bar{x}$  a média da amostra

e  $\sigma$  o desvio padrão da amostra.

Após a predição vamos calcular o erro médio e erro absoluto:

O erro médio diz em média o quanto o modelo erra, o erro absoluto diz, proporcionalmente à média o quanto o modelo erra:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

$$MAPE = \frac{MAE}{\bar{Y}}$$

Por motivos de leitura, calcularemos:  
1- MAPE = O quanto o modelo acerta em torno da média. Do qual o resultado foi surprendentemente positivo:  $1-MAPE \approx 0,6$  ou

**60%** de acertos em torno da média. Esse resultado é mais que ótimo para preenchermos os dados faltantes.

Agora que estamos prontos para calcular o PCA, os primeiros  $k$  elementos são os autovetores da matriz de correlação (no nosso caso, pela padronização):

$PV = \lambda V$ ,  $\lambda$  indica quanta correlação a combinação do PC k capta.

Para a regressão do PCR precisamos ajustar uma logística:

A ideia é o modelo devolver um probabilidade baseada na variável target binária, de formato bernoulli:

$P(Y|X) = \beta_0 + \beta_1 X$ , porém  $\beta_0 + \beta_1 X$  não está no domínio  $[0, 1]$ , logo, podemos usar a função logística:

$$p(x) = P(Y|X) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

Agora, como conhecemos a f.d.p de  $p(x)$   
podemos estimar usando máxima verossimilhança

$$p(x) \sim \text{bernhoulli}(p) \quad \text{com } p = \frac{e^h}{1+e^h}$$

$$h: \beta_0 + \beta_1 x$$

Logo, queremos estimar a função

$$f(y|p) = p^y (1-p)^{1-y}, y = \begin{cases} 1, \text{ se sim} \\ 0, \text{ não} \end{cases}$$

$$L(x) = \prod_{i=1}^n p^{y_i} (1-p)^{1-y_i}$$

$$l(x) = \sum_{i=1}^n \log p^{y_i} + \log (1-p)^{1-y_i}$$

$$= \sum_{i=1}^n y_i \log p + (1-y_i) \log (1-p)$$

$$\text{Agora } p = \frac{e^h}{1+e^h} \Rightarrow 1+e^h = e^{-h} \Rightarrow r = \frac{e^h}{1+e^h} = \frac{e^h}{e^{-h}} = e^h$$

$$\Rightarrow 1 - \frac{e^h}{1+e^h} = p e^h \Rightarrow r = \frac{e^h}{1-p}$$

$$\Rightarrow \frac{1}{e^h} : \frac{1-p}{r} \Rightarrow e^h = \frac{p}{1-p}, \text{ logo} \quad \frac{1+e^h - e^h}{1+e^h} = \frac{1}{1+e^h}$$

$$\therefore \sum_{i=1}^n y_i \left( \log \left( \frac{e^h}{1+e^h} \right) \right) + (1-y_i) \log \left( 1 - \frac{e^h}{1+e^h} \right)$$

$$\Rightarrow \sum_{i=1}^n y_i (\log e^h - \log (1+e^h)) + (1-y_i) \log \left( \frac{1}{1+e^h} \right)$$

$$\Rightarrow \sum_{i=1}^n y_i (h - \log (1+e^h)) + (1-y_i) \log \left( \frac{1}{1+e^h} \right)$$

$$\sum_{i=1}^n y_i \left( h - \log(1+e^h) \right) + (1-y_i) \left( \log 1 - \log(1+e^h) \right)$$

$$\rightarrow \sum_{i=1}^n y_i \left( h - \log(1+e^h) \right) + (1-y_i) \left( -\log(1+e^h) \right)$$

$$\rightarrow \sum_{i=1}^n y_i h - y_i \cancel{\log(1+e^h)} - \log(1+e^h) + y_i \cancel{\log(1+e^h)}$$

$$\rightarrow \sum_{i=1}^n y_i h - \log(1+e^h)$$

$$\rightarrow \sum_{i=1}^n y_i (\beta_0 + \beta_1 x_i) - \log \left( 1 + e^{\beta_0 + \beta_1 x_i} \right)$$

Derivando igualando a 0

$$\frac{\partial l(\beta_0, \beta_1)}{\partial \beta_0} = 0, \quad \frac{\partial l(\beta_0, \beta_1)}{\partial \beta_1} = 0$$

$$\begin{cases} \frac{\partial l}{\partial \beta_0} \\ \frac{\partial l}{\partial \beta_1} \end{cases} \rightarrow \sum_{i=1}^n y_i - \frac{1}{1 + e^{\beta_0 + \beta_1 x_i}} \cdot e^{\beta_0 + \beta_1 x_i}$$

$$\rightarrow \sum_{i=1}^n y_i - \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}}$$

$$\rightarrow \sum_{i=1}^n y_i - p_i = 0 \rightarrow \beta_0$$

$$\frac{\partial l}{\partial \beta_1}$$

$$\rightarrow \sum_{i=1}^n y_i x_i - \frac{1}{1 + e^{\beta_0 + \beta_1 x_i}} \cdot e^{\beta_0 + \beta_1 x_i} \cdot x_i$$

$$\Rightarrow \sum_{i=1}^n y_i x_i = \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}} \cdot x_i$$

$$\Rightarrow \sum_{i=1}^n y_i x_i - p_i x_i \rightarrow \sum_{i=1}^n x_i (y_i - p_i)$$

$$\Rightarrow \sum_{i=1}^n x_i (y_i - p_i) = 0 \Rightarrow \beta_1$$

Por fim:  $\beta_0 \Rightarrow \sum_{i=1}^n y_i - p_i = 0$

$\beta_1 \Rightarrow \sum_{i=1}^n x_i (y_i - p_i) = 0$

Ou seja, em forma matricial o resultado é:

$$\nabla l(\beta) = X^T (y - p) = 0$$

$$\text{Com } p = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

Para o cálculo numérico, vamos usar o gradiente ascendente:

$$\beta^{(t+1)} = \beta^{(t)} + \lambda \nabla f(\beta)$$

Com  $\lambda$  sendo chamado de learn rate, basicamente a função usa como peso: menor chance de passar na direção

que venha a diminuir o valor da função que minimize  $\beta$ . Com  $n$  épocas, que serão quantos passos o modelo dará em direção ao gradiente.

Com isso, basicamente todas funções e matemática estarão pronta.

Score: 0,74401