COMO FAZER REGRESSÃO LINEAR EM PYTHON SEM PACOTES

Armando Handaya[1]

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – IFSP

Resumo
Este minicurso abordará a técnica de Regressão Linear usando a linguagem python e a plataforma Colab do Google, mas sem apelar para os diversos pacotes específicos existentes de Aprendizado de Máquina, Inteligência Artificial ou outro qualquer, que executam essa tarefa de Regressão Linear automaticamente. Esses pacotes específicos, que fornecem o resultado de imediato, são bons para aqueles usuários que não se importam em saber como foi feita a conta, como foi o processo para chegar aonde chegou. Neste minicurso explicaremos esse processo detalhadamente e fundamentada em bases matemáticas precisas, usando o cálculo matricial. A abordagem teórica será feita vetorialmente, mas as contas serão executadas matricialmente, o que deve encurtar todo o processo. Para isso vamos usar um pacote genérico de manipulação matricial, que é o *sympy.matrices*. Dessa forma no título deste minicurso o termo "sem pacote" quer dizer sem pacote específico.
Palavras-chave: Regressão Linear; Cálculo Numérico; Cálculo Vetorial; Cálculo Matricial; Aprendizado de Máquina.

1. INTRODUÇÃO

A regressão linear é uma técnica estatística fundamental para modelar e entender a relação entre variáveis. Ela é amplamente utilizada em diversas áreas, desde finanças e economia até ciência dos dados e pesquisa científica. Este minicurso tem como objetivo fornecer uma introdução abrangente à regressão linear, demonstrando como aplicá-la utilizando a linguagem de programação Python, uma ferramenta poderosa e popular na comunidade de ciência de dados.

Ao longo deste curso, exploraremos os princípios básicos da regressão linear, discutindo seus fundamentos matemáticos e como implementá-la em Python sem depender de pacotes específicos. Esta abordagem permitirá que você compreenda

[1]Doutor em Engenharia Elétrica pela Universidade de São Paulo (USP). Professor do EBTT do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP), Guarulhos, SP, Brasil. E-mail: ahand@ifsp.edu.br.

profundamente o funcionamento da regressão linear, desde a formulação do modelo até a interpretação dos resultados.

1.1 Por que Regressão Linear?

A regressão linear é uma técnica simples, mas poderosa, para modelar a relação entre uma variável dependente (ou resposta) e uma ou mais variáveis independentes (ou preditoras). Ela é frequentemente utilizada para prever ou explicar o comportamento de uma variável com base em outras variáveis explicativas. Por exemplo, em um contexto financeiro, pode ser usada para prever o preço de uma ação com base em fatores como lucros da empresa, volume de negociações e indicadores econômicos.

Além de sua simplicidade, a regressão linear oferece interpretações intuitivas dos resultados. Os coeficientes do modelo podem ser diretamente interpretados como o impacto médio de uma unidade de mudança na variável independente sobre a variável dependente, mantendo todas as outras variáveis constantes. Isso torna a regressão linear uma ferramenta valiosa para análise e tomada de decisões em uma variedade de contextos.

1.2 Por que Python?

Python é uma linguagem de programação de propósito geral que se tornou extremamente popular na comunidade de ciência de dados e aprendizado de máquina. Sua sintaxe relativamente simples e legível, juntamente com uma ampla gama de bibliotecas e ferramentas, tornam-na uma escolha ideal para análise de dados e modelagem estatística.

Ao optar por implementar a regressão linear em Python sem depender de pacotes específicos, você terá a oportunidade de compreender completamente o funcionamento interno da técnica. Isso inclui a formulação matemática do modelo, a implementação dos algoritmos de ajuste de parâmetros e a interpretação dos resultados. Embora bibliotecas como NumPy, SciPy e scikit-learn ofereçam implementações eficientes da regressão linear, construir o modelo a partir do zero em Python puro proporcionará uma compreensão mais profunda de seus princípios subjacentes.

1.3 Estrutura do Minicurso

Este minicurso será dividido em várias seções, cada uma cobrindo um aspecto fundamental da regressão linear:

1. Introdução à regressão linear e formulação do modelo.
2. Método dos mínimos quadrados para estimativa dos parâmetros.
3. Implementação do modelo de regressão linear em Python sem pacotes específicos.

4. Avaliação do modelo e interpretação dos resultados.

5. Extensões e considerações adicionais na regressão linear.

Cada seção será acompanhada de exemplos práticos e exercícios para reforçar os conceitos apresentados. Ao final do minicurso, você terá uma compreensão sólida da regressão linear e será capaz de aplicar esse conhecimento em seus próprios projetos de análise de dados usando Python.

2. FUNDAMENTAÇÃO TEÓRICA

A forma mais simples de regressão linear, conhecida como regressão linear simples, envolve apenas uma variável independente, enquanto a regressão linear múltipla permite a inclusão de múltiplas variáveis independentes.

A formulação matemática da regressão linear é representada pela equação:

$$Y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_n x_n + \varepsilon$$

onde:

- Y é a variável dependente;

- α_o é o intercepto, que representa o valor de Y quando todas as variáveis independentes são iguais a zero;

- $\alpha_1, \alpha_2, \ldots \alpha_n$ são os coeficientes de regressão, que representam o impacto de cada variável independente em Y;

- $x_1, x_2, \ldots x$ são as variáveis independentes;

- ε é o termo de erro, que captura a variação não explicada pelo modelo.

O objetivo da regressão linear é estimar os coeficientes $\alpha_o, \alpha_1, \alpha_2, \ldots, \alpha_n$ que minimizam a soma dos quadrados dos resíduos (RSS), também conhecida como método dos mínimos quadrados. Este método busca encontrar os melhores parâmetros para o modelo, de modo a minimizar a diferença entre os valores observados e os valores preditos pelo modelo.

A estimativa dos coeficientes pode ser obtida através de várias abordagens, sendo a mais comum a utilização do método dos mínimos quadrados ordinários (OLS). Este método calcula os coeficientes de forma analítica, encontrando os valores que minimizam a função de custo RSS. A melhor reta da regressão linear, evidentemente, é aquela cujo erro é igual a zero, mas isso quase nunca acontece. Buscando assumir esse ideal a equação que precisamos trabalhar, restringindo para $n = 1$, é a seguinte

$$Y = \alpha_0 + \alpha_1 x$$

ou em formato matricial, e assumindo as coordenadas em R^{n+1}.

$$\underbrace{\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{\vec{y}} = \underbrace{\begin{pmatrix} \vec{1} & \vec{x} \end{pmatrix}}_{M} \cdot \underbrace{\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}}_{\vec{a}}$$

Como o nosso objetivo são os coeficientes α_i ou seja o vetor a, matricialmente basta isolá-lo na equação acima. Para isso a matriz *M* precisa ser repassada para o outro lado.

Agora, para fazer isso, a matriz *M* precisa ser quadrada e invertível, o que de fato ela geralmente não é (só é quadrada quando $n = 1$). Para contornar esse problema, vamos trabalhar com uma outra equação:

$$M^t y = M^t M . a$$

onde a matriz $M^t M$ é quadrada e praticamente invertível sempre. Dessa forma podemos isolar o vetor a:

$$a = (M^t M)^{-1} M^t y$$

Com essa fórmula obtemos os coeficientes da reta regressão em questão. Os cálculos serão feitos exatamente dessa forma, matricialmente.

3. MEDINDO A QUALIDADE DO AJUSTE

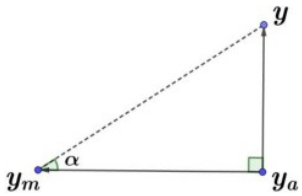
Após estimar os coeficientes do modelo, podemos avaliar sua qualidade ajustando-o aos dados observados. Uma medida comum de ajuste é o coeficiente de determinação (r^2), que varia de 0 a 1 e indica a proporção da variabilidade da variável dependente que é explicada pelo modelo. Quanto mais próximo de 1, melhor é o ajuste do modelo aos dados.

Depois de calcular o vetor *a*, o que obtemos ao fazer a multiplicação $M \cdot a$ não será exatamente o vetor *y*, mas a melhor aproximação dele no plano gerado pelos vetores das coordenadas de *M*. A esse vetor chamamos de

$$y_a = M . a$$

Considerando um terceiro vetor y_m obtido de y trocando cada uma das suas coordenadas pela média, ou seja $y_m = \bar{y} . 1$

esses três vetores formam um triângulo retângulo



e o cosseno do ângulo α fornece o coeficiente da correlação linear, sem o sinal. O coeficiente de determinação é dado por

$$r^2 = cos^2\alpha$$

e ele é usado para medir a qualidade do ajuste da reta da regressão.

4. IMPLEMENTAÇÃO EM PYTHON

Implementar a regressão linear em Python sem depender de pacotes específicos envolve traduzir esses conceitos matemáticos em código Python. Isso inclui a leitura dos dados, a construção da matriz de design *M*, o cálculo dos coeficientes de regressão usando o método dos mínimos quadrados e a avaliação do modelo. Embora bibliotecas como NumPy e SciPy ofereçam funções que facilitam esses cálculos, entender como implementar a regressão linear do zero em Python proporciona uma compreensão mais profunda de seus fundamentos teóricos e práticos.

Os passos da implementação são:

1. Importar os comandos necessários

from *sympy.matrices* import *Matrix*, *ones*

2. Definir as coordenadas dos vetores ***x***, ***y*** e **1** (uns)

$$x = Matrix[(x_0, x_1, \dots, x_n)]$$

$$y = Matrix[(y, y_1, \dots, y_n)]$$

$$uns = ones(len(x), 1)$$

3. Criar as matrizes dos coeficientes e sua transposta

M=uns.row_join (x)

N = M.T

4. Inserir a fórmula

a = (N*M)**-1 *N*y

5. Enfim criar a reta da regressão linear

F = lambda x : a0+a1*x

onde a0 e a1 são as coordenadas do vetor a: a0=a[0] e a1=a[1].

6. Para fazer a estimativa para novos dados (=data) podemos usar o seguinte comando, depois de definir o vetor data = [num1, num2, …]

for i in data: print(F(i))

7. Para calcular o coeficiente de determinação (score) podemos usar a seguinte sequência de comandos

```
from numpy import mean,
sign ya = M*a
ym =
mean(y)*uns u =
ya-ym
v = y-ym
cos =
u.norm()/v.norm()
f'{cos**2:.5f}'
```

Para completar a informação sobre o coeficiente de correlação linear r, basta considerar o sinal de $a1$, ou seja $r = sinal(a1)\, cos$. A última linha nas linhas de código acima é o score ou coeficiente de determinação, cos**2, que fornece a qualidade do ajuste da regressão linear. Demais comando dessa linha fornece o formato de saída numérica com 5 casas decimais.

Para fazer o gráfico da reta regressão linear, é possível usar o python, porém, o jeito mais fácil é utilizar o software Geogebra, pois nele não há a necessidade de definir o eixo *x*, nas apenas os valores da função *y=f(x)*.

REFERÊNCIAS

DECHEN, A.R. **Análise de Regressão: Modelagem Quantitativa da Relação entre Variáveis**. São Paulo: Oficina de Textos, 2020.

GÉRON, D. **Machine Learning: This Book Includes: Machine Learning for Beginners, Machine Learning with Python**. 1º Ed., Sem editor (Publicação Independente), 2019.

GUERRA, R.A.M. **Análise de Regressão: Uma Abordagem Moderna**. Rio de Janeiro: LTC, 2017.

HANDAYA A. Uso de Tecnologia em Sala de Aula: um Relato de Experiência no Instituto Federal de São Paulo câmpus Guarulhos. In: Luís Eduardo Maggi. (Org.). **Pesquisas no Ensino Básico, Técnico e Tecnológico: Ciências Exatas**. 1ed.Rio Branco: Stricto Sensu Editora, 2020, v. 3, p. 209-226.

MULLER, A. C.; GUIDO, S. **Introduction to Machin.e Learning with Python: A Guide for Data Scientists**. 1ºEd, Editora O'Reilly, 2016

SCHIMIGUEL, J.S.; MAGALHÃES, M.A.B. **Análise de Regressão Linear: Um Guia Prático com Aplicações em R**. Rio de Janeiro: Elsevier, 2018.