#867 #DesenvolveDados Seg • Qua • Sex

Estatística II

Conteúdo

2 Regressão Linear

Regressão Linear

1. Introdução

As **Regressões** são um tipo de modelo de *Machine Learning*, no caso onde a aprendizagem é **supervisionada**. Ou seja, para o funcionamento do modelo serão fornecidas informações na forma de variáveis atributos (também nomeadas de *features* de um conjunto de dados) e o modelo estimará o valor da variável resposta (*target*) usando dados de referência durante o treinamento. Lembrando que, para o caso da Regressão, o tipo de resposta esperado na saída será um **valor contínuo** representativo de acordo com a variável resposta.

Especificamente quando se trata de **Regressões Lineares**, a inferência feita sobre a relação entre as variáveis é que pode ser descrita por uma **equação de reta**.

2. Regressão Linear Simples

Na regressão linear simples, tem-se um conjunto de dados formado por um único atributo X e a variável resposta Y. O modelo vai procurar estabelecer a melhor equação de reta que descreva o conjunto de dados, ou seja define a equação como:

$$Y \approx \beta_0 + \beta_1 X$$

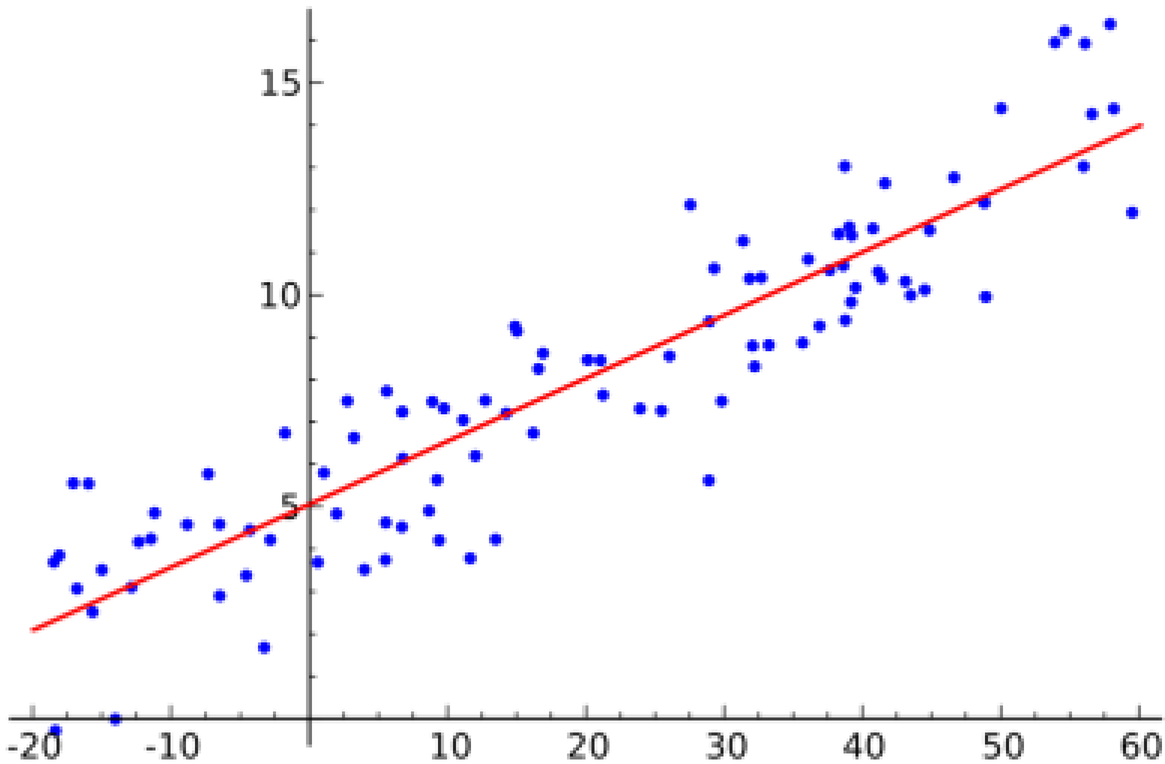
Uma forma para estimar os valores de β_0 e β_1 é a partir dos **valores médios** para X e Y, onde β_0 é o **coeficiente independente** (onde a equação de reta vai cortar o eixo Y no gráfico) e o β_1 é o **coeficiente angular** desta reta (indicando a inclinação da reta a ser ajustada). Assim os valores dos coeficientes também serão um valor médio da forma $\hat{\beta}_0$ e $\hat{\beta}_1$, dados pelas seguintes equações:

$$\hat{\beta}*1=\frac{\sum*{i=1}^n (x_i- \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{\sigma_{xy}}{\sigma_{xx}} = \frac{Covar(x, y)}{Var(x)}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

Onde:

- **Covariância** ou variância conjunta, que indica o grau de interdependência entre duas variáveis;
- **Variância** é uma medida de o quão disperso estão os dados, ou seja o quão distante está cada valor desse conjunto do valor médio.



Fonte: Imgur

Uma implementação dessas equações para o cálculo dos coeficientes utilizando o *Python* e algumas bibliotecas auxiliares pode ser feita seguindo o exemplo abaixo:

```
def linear_regression(x, y):
    # Definir as médias
    mean_x, mean_y = np.mean(x), np.mean(y)

    # Calcular a covariância e variância
    S_xy = 0
    S_xx = 0

    #Laço para o somatório
    for i in range(0, len(x)):
        # termo covariância
```

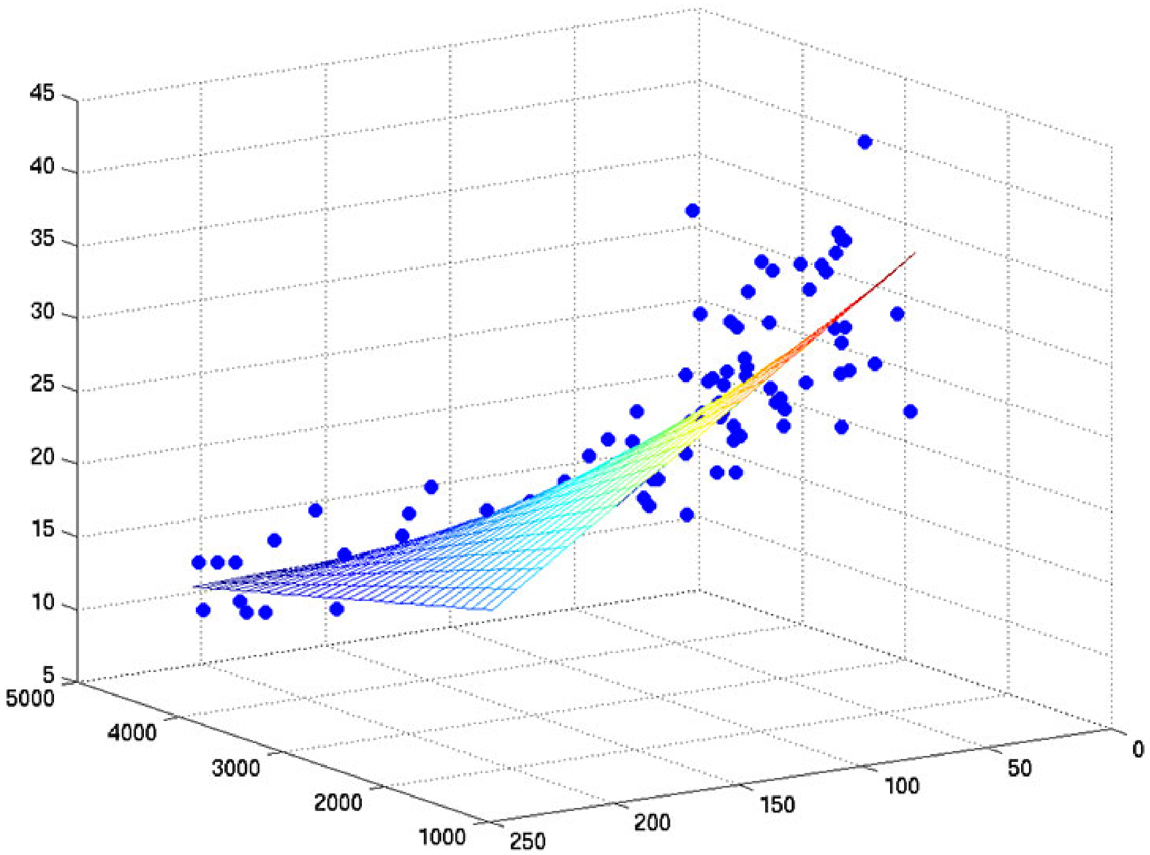
```
        S_xy += (x[i] - mean_x)*(y[i] - mean_y)
        # termo variância
        S_xx += (x[i] - mean_x)**2

    # Calcular os coeficientes de regressão
    beta_1 = S_xy / S_xx
    beta_0 = mean_y - beta_1*mean_x

    # Retorna os coeficientes
    return beta_0, beta_1
```

3. Regressão Linear Múltipla

A Regressão Linear Múltipla seria uma generalização da Regressão Linear Simples, onde agora trabalha se com n variáveis atributo X e uma variável resposta Y. Dessa forma, a equação que descreve o modelo para Regressão Linear Múltipla é descrita conforme a seguir:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n$$

De forma análoga a Regressão Linear Simples, o termo β_0 será o coeficiente independente desta equação de reta e os termos $\beta_1, \beta_2, ..., \beta_n$ serão os coeficientes angulares das respectivas variáveis $X_1, X_2, ..., X_n$. Note que agora, ao ajustar uma Regressão Linear Múltipla, procura-se determinar um **hiperplano** de dimensão n que melhor se ajuste ao conjunto de dados. Sobre a aplicabilidade dos modelo de Regressão Linear vale ressaltar uma condição para ser válida a aplicação desta técnica, que seria a **independência das observações**, ou seja as observações não têm relações diretas com as demais e não são de forma sequencial (como no caso de **Séries Temporais**).



Fonte: Oficina da Mente

Para utilizar uma Regressão Linear Múltipla, será necessário carregar a função oriunda do *Scikit-Learn*, conforme indicado a seguir:

```
from sklearn.linear_model import LinearRegression
```

Após carregado a função para Regressão Linear no ambiente, processo para modelagem vai seguir em 3 passos:

- **Instanciar o modelo**: Processo onde será construído o objeto com as funções da Regressão Linear;
- **Treinamento do Modelo**: Passo onde são passados dados de treinamento com a respectiva resposta, para o modelo aprender a generalizar e chegar o mais póximo das resposta;
- **Gerar novas predições**: Com o modelo já ajustado e treinado, o último passo seria gerar novas predições para outros dados (comumente gera-se predições para dados de teste para poder avaliar o desempenho do modelo).

A implementação destes tópicos pode ser feita seguindo o código abaixo:

```
# Instanciar o modelo
model = LinearRegression()
```

```
# Treinamento do Modelo
model.fit(X_train, y_train)

# Gerar novas predições
y_pred = model.predict(X_test)
```

Após treinar o modelo, para acessar os valores dos coeficientes $\beta_0, \beta_1, ..., \beta_n$, utiliza-se de duas funções do próprio modelo, conforme mostrado abaixo:

```
# Determina o valor de beta_0 (valor que intercepta o eixo y)
print(model.intercept_)

# Determinando os demais coeficientes betas
print(model.coef_)
```

4. Resíduos

Os resíduos ou valor residual de uma Regressão Linear qualquer é definido como a diferença entre o valor real de uma determinada observação e o valor predito pelo o modelo, dado pela seguinte fórmula:

$$Re(y_i) = y_i - \hat{y_i}$$

Uma boa indicação da qualidade do ajuste de uma Regressão Linear a um conjunto de dados é que a **distribuição dos resíduos** siga uma **distribuição normal**.

Materiais Complementares

Canal *StatsQuest*, vídeo sobre Linear Regression Clearly explained;

Documentação no Scikit-Learn sobre o Linear Regression;

Artigo publicado por Abhigyanl no *Analytics Vidhya* - *Understanding The Linear Regression!!!!*.

Tópico anterior

Próximo Tópico