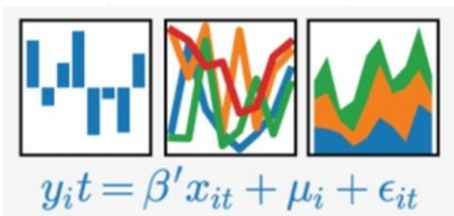
@Odemir Depieri Jr

Guia de Machine Learning - Regressão Linear Simples

Python



@Odemir Depieri Jr

Resumo sobre Regressão Linear

A analise de **regressão linear** avalia a **relação estatística** entre uma ou mais variáveis. Vamos exemplificar ...

Variação de gasto com **Alimentação** x **Salário**

Concessão **limite** de cartão de crédito x **Salário**

Quando consideramos o efeito de duas ou mais variáveis, utilizamos a analise de **regressão múltipla**, exemplo:

Alimentação Salário Limite Crédito

Quando consideramos o efeito de apenas 1 variável, utilizamos a analise de **regressão simples**.

Para que serve uma regressão ?

O modelo de **regressão** serve **para prever** comportamentos com base na associação entre duas variáveis que geralmente possuem uma boa correlação.

Onde utilizo essa regressão ?

As aplicações são diversas !! Mas vamos exemplificar:

- ✓ **Prever o valor** de fechamento de uma ação na ibovespa;
- ✓ **Produtividade** de colaboradores de um call center;
- ✓ **%** de Desmatamento nos próximos anos;
- ✓ **Previsão** de faturamento.
- ✓ Muitos outros !

Observação: Estatística **não** é 100% assertiva, **o papel dela é te direcionar.**

@Odemir Depieri Jr

Regressão Linear - Simples

Vamos importar as bibliotecas externas que iremos precisar

```
[18] # Biblioteca para modelagem de dados
     import pandas as pd

     # Biblioteca para recursos matemáticos
     import numpy as np

     # Biblioteca para recursos Graficos
     import matplotlib.pyplot as plt
```

Vamos criar alguns dados fictícios para podemos usar nesse exemplo

```
[40] # Criando nossa base de dados
     # ---------------- Base de Preço de apartamentos Fictícia ----------

     # Criando lista com os valores
     Metragem = [40, 45, 50, 55, 60, 62, 65, 70, 80, 90, 92,
                 100, 110, 120, 150]
     Valor = [200, 280, 310, 350, 390, 410, 450, 490, 550, 620,
              670, 700, 750, 810, 989]

     # Organizando os valores em um Dicionário
     Dicionario = {
         'Metragem' : Metragem,
         'Valor Imovel' : Valor
     }

     # Lendo o Dicionário com o Pandas
     DataFrame = pd.DataFrame( data=Dicionario )

     # Verificando as primeiras linhas
     DataFrame.head()
```

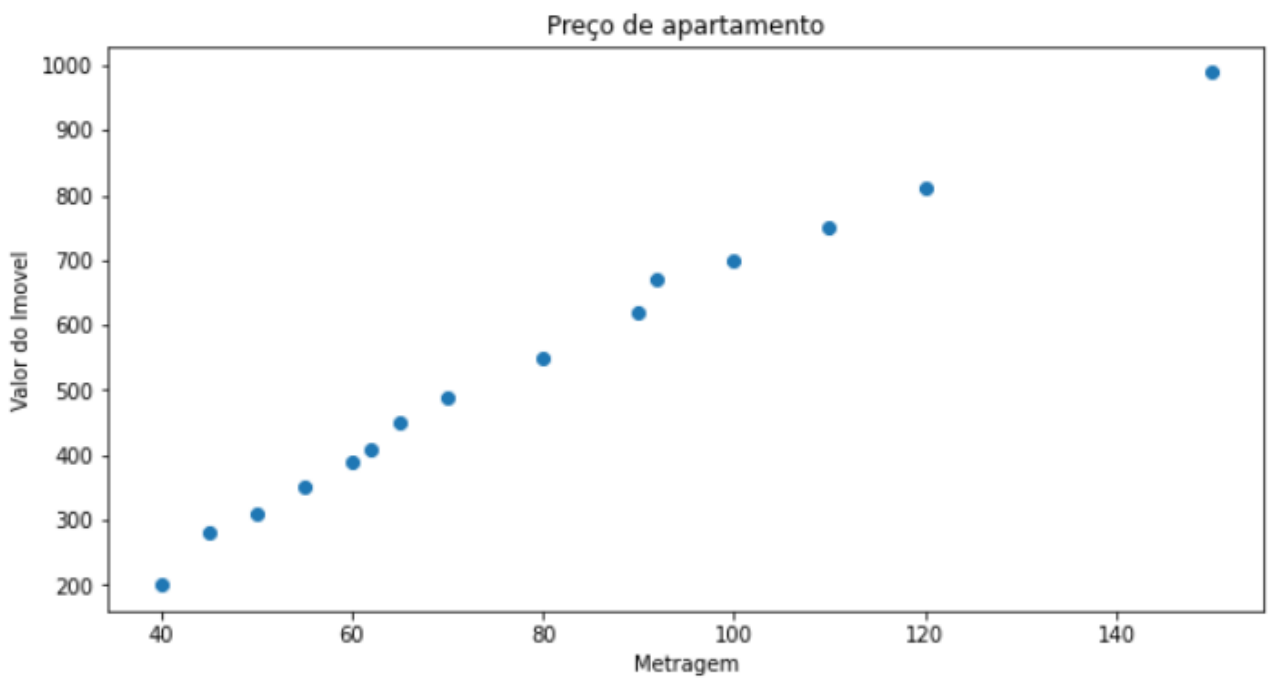
	Metragem	Valor Imovel
0	40	200
1	45	280
2	50	310
3	55	350
4	60	390

Vamos usar esses dados para prever o **valor de um apartamento**.
Nesse exemplo criamos alguns dados fictícios para treinamos o modelo.

Regressão Linear - Simples

Vamos gerar um gráfico com 2 eixos para entender os dados

```
[42]  # -------------- Plotagem dos dados --------------

      # Ajustando o tamanho do Gráfico
      plt.figure( figsize=(10,5) )
      # Passando os valores para o grafico
      plt.scatter( DataFrame['Metragem'].values,
                   DataFrame['Valor Imovel'].values )
      # Definindo um titulo
      plt.title('Preço de apartamento')
      # Definindo o nome do eixo x
      plt.xlabel('Metragem')
      # Definindo o nome do eixo y
      plt.ylabel('Valor do Imovel');
```



Nesse exemplo os dados foram forjados.
É nítido que há uma correlação entre essas 2 variável.
Quanto **maior a metragem** maior o **valor do imóvel**.

Vamos calcular a correlação

```
[43] # Separando os dados no eixo x e y
     Eixo_x = DataFrame.iloc[:,0].values
     Eixo_y = DataFrame.iloc[:,1].values

     # Calculando a correlação entre os dados usando o Numpy
     Correlacao = np.corrcoef( Eixo_x, Eixo_y )
     Correlacao
```

```
array([[1.        , 0.99320395],
       [0.99320395, 1.        ]])
```

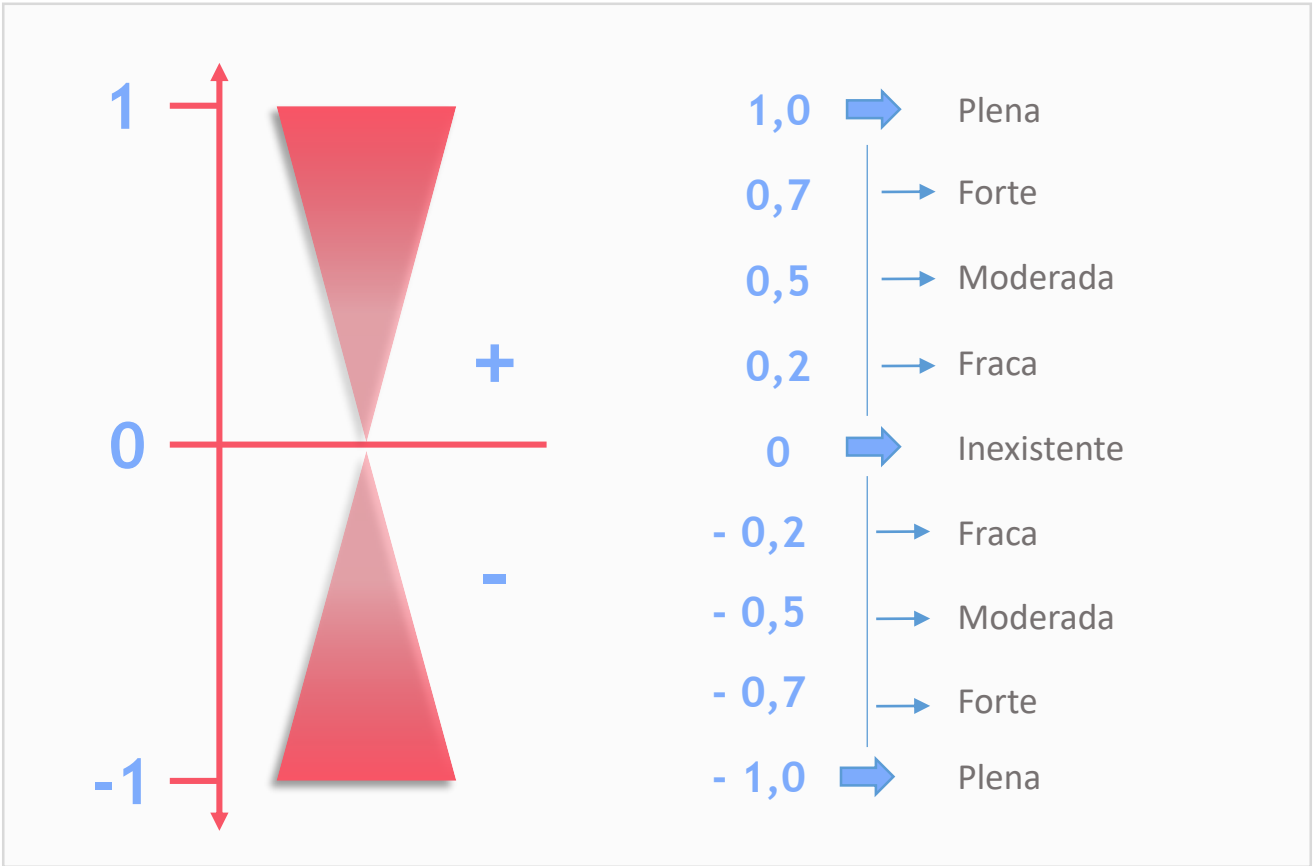
A correlação das variáveis ficou em 0.99 (Forte)
Verifique na tabela abaixo a escala.

@Odemir Depieri Jr

Regressão Linear - Simples

Vamos entender a tabela de correlação

Tabela de Correlação



A correlação pode ser **positiva** ou **negativa**, a escala vai de **1** a **-1**.

Quanto mais próximo de 1, há uma **correlação positiva,** ou seja, quando uma variável cresce a outra cresce.

Quanto mais próxima de -1, há uma correlação negativa, ou seja, quando uma variável cresce a outra diminui ou vice-versa.

No **nosso exemplo** a correlação ficou em 0.99, ou seja, há uma correlação muito forte entre o metragem vs preço.

```
array([[1.        , 0.99320395],
       [0.99320395, 1.        ]])
```

Na regressão é **sempre importante haver correlações fortes** entre as variáveis.

Caso não haja, o modelo irá ter uma **dispersão muito grande**, e as previsões ficaram fora da realidade.

Regressão Linear - Simples

Ajustar os dados em uma matriz para serem inseridos no 'sklearn'

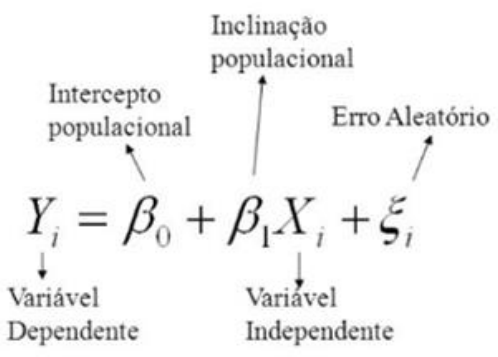
```
[44] # Convertendo o Eixo x para formato de Matriz
     # -1 quer dizer para não mexer nas linhas,
     # 1 quer dizer para incluir uma coluna

     Eixo_x = Eixo_x.reshape(-1, 1)
```

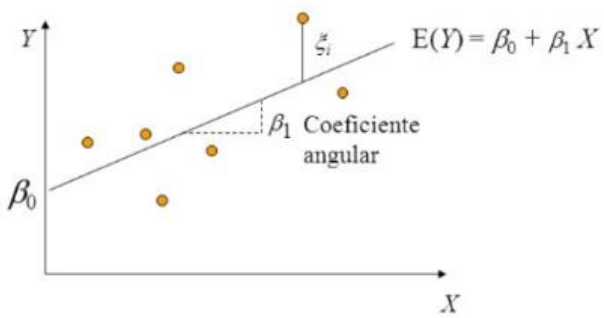
Vamos treinar o modelo

```
[51] # Importando a biblioteca com os recursos da regressão linear
     from sklearn.linear_model import LinearRegression

     # Definido uma variavel com os calculos estatisticos
     Regressor = LinearRegression()

     # Passando os dados para treinar o modelo
     Regressor.fit( Eixo_x, Eixo_y )

     # Identificando o b0 e b1 - Referencia da formula da regressão line
     B1 = Regressor.coef_ # Coeficiente
     B0 = Regressor.intercept_ # Constante

     # Mostrando os valores
     print('B0 =', B0, '\n', 'B1 =', B1[0] )
```

```
B0 = -32.38752208173719
 B1 = 7.110860244933606
```

Formula da regressão linear





Caso não tenha conhecimento sobre estatística, vou deixar um vídeo bem legal de como funciona essa formula.

https://www.youtube.com/watch?v=n--K70T6c3A&list=PLVGJZxcisYSGmqs4muOxEoiqUYWlAVmef&index=16

Regressão Linear - Simples

Vamos Calcular o score dessa regressão.

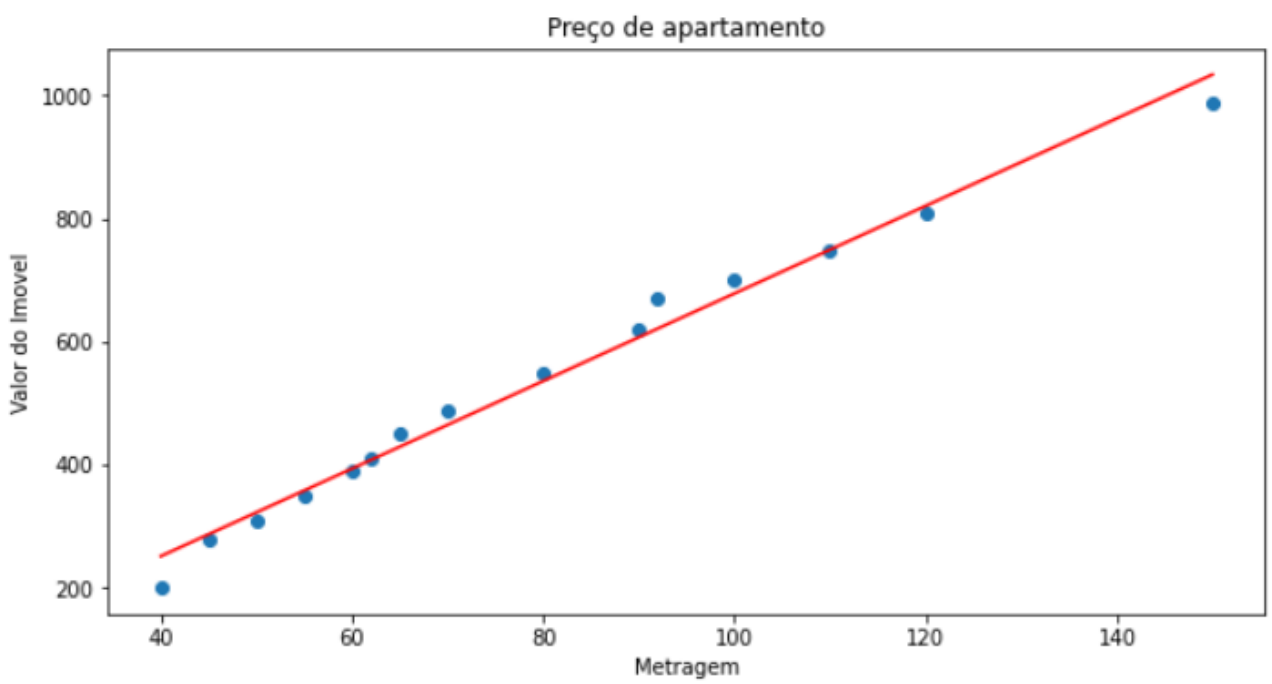
```
[52] # Calculando o score da regressão
     Score = Regressor.score( Eixo_x, Eixo_y )
     Score
```

```
     0.9864540901286758
```

Esse score é diferente da correlação.
Porem a escala é bem similar, quanto mais próximo de 1 é melhor.

Vamos entender a reta que foi gerada pelo modelo

```
[47]  # -------------- Plotagem da Reta --------------

     # Ajustando o tamanho do Gráfico
     plt.figure( figsize=(10,5) )
     # Passando os valores para o grafico
     plt.scatter( DataFrame['Metragem'].values,
                  DataFrame['Valor Imovel'].values )
     # Plotando a reta gerada pela regressão
     plt.plot( Eixo_x, Regressor.predict(Eixo_x), color='red' )
     # Definindo um titulo
     plt.title('Preço de apartamento')
     # Definindo o nome do eixo x
     plt.xlabel('Metragem')
     # Definindo o nome do eixo y
     plt.ylabel('Valor do Imovel');
```



Avaliando o modelo.
Nessa previsão a **reta vermelha** ficou bem próxima dos dados, ou seja, o modelo teve uma ótima performance.
Lembrando que os dados são forjados. Nem sempre ficara dessa forma nos modelos rsrs.

Regressão Linear - Simples

Fazendo uma previsão

```
[66] # ------- Fazendo previsões -------
     # Vamos prever o valor de um apartamento de 50 metros
     Qual_Tamanho_Apartamento = [[ 50 ]]

     # Prevendo quanto custaria o valor de um apartamento
     Previsão = Regressor.predict( Qual_Tamanho_Apartamento )

     print('Um apartamento de:',
           Qual_Tamanho_Apartamento[0][0], 'metros' )
     print('Usando o modelo para prever o valor, custaria: R$',
           round(Previsão[0],2))
```

```
Um apartamento de: 50 metros
Usando o modelo para prever o valor, custaria: R$ 323.16
```

Nesse exemplo acima, fizemos uma previsão de um valor de uma apartamento de 50 metros.

Nos nossos dados um aparamento de 50 metros custava 310.
O modelo previu 323.

Ressalto que estatística não é 100% assertiva. Ela é diretiva.

Final

Esse guia é um exemplo de uma regressão linear simples.

Guia da documentação caso queira mais detalhes
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

Odemir Depieri Jr

Software Engineer Sr
Tech Lead
Specialization AI