

**1** Aqui temos uma base de dados com dois atributos.  
Os quilômetros rodados e o valor cobrado por  
conta daqueles quilômetros.

**3**

Base de treino: composta, nesse  
caso, por 5 valores da base  
(~70%), ela vai ser usada para a  
criação (ou treino) do modelo.

Km rodados	Valor cobrado
2	5
3	7
4	9
6	13
4	9
40	81
5	11

**2**

Sendo assim, a ideia é criar um modelo que  
represente esses dados para, quando  
chegarem novos valores de quilômetros  
rodados, o valor seja calculado previamente  
(para mostrar esses valores para o cliente de  
um aplicativo de corrida, por exemplo).

**4**

Base de teste: composta, nesse  
caso, por 2 valores da base  
(~30%), ela vai ser usada para a  
avaliação (ou teste) do modelo.

**Para a execução dos comandos presentes nas páginas a seguir você vai usar as seguintes bibliotecas (copie as linhas a seguir no seu código):**

```
from sklearn.linear_model import LinearRegression  
import pandas as pd
```

**Para criar a base de dados de exemplo, use o seguinte comando:**

```
df = pd.DataFrame([[2,3,4,6,4,40,5],[5,7,9,13,9,81,11]]).T  
df.columns = ['Km rodados','Valor cobrado']
```

**1** Definimos como X a base com as variáveis independentes e y a base com variável independente.

**2** A definição da base de treino é feita pelos comando:

```
X_train = df[['Km rodados']].iloc[:5]  
y_train = df['Valor cobrado'].iloc[:5]
```

Km rodados	Valor cobrado
2	5
3	7
4	9
6	13
4	9
40	81
5	11

**3** A definição da base de teste é feita pelos comandos:

```
X_test = df[['Km rodados']].iloc[5:]  
y_test = df['Valor cobrado'].iloc[5:]
```

1

Com as bases definidas, agora devemos usar elas para o devido propósito. No caso da base de treino, vamos treinar o modelo.

2

Primeiro, vamos criar uma variável que vai ser usada para executar a regressão. Ela pode ser chamada de reg:

```
reg = LinearRegression()
```

3

Depois, usamos a variável que contém a regressão para criar (treinar) o modelo:

```
reg.fit(X_train, y_train)
```

Km rodados	Valor cobrado
2	5
3	7
4	9
6	13
4	9
40	81
5	11

4

Agora temos um modelo, em forma de regressão linear, chamado aqui de reg:

reg





Portanto, os dados apresentados na base de treino foram usados por um algoritmo chamado Método dos Mínimos Quadrado para gerar uma regressão linear, chamada aqui de reg.

Km rodados	Valor cobrado
2	5
3	7
4	9
6	13
4	9
40	81
5	11



reg ->  $y = 2X + 1$

1

Com o modelo criado, vamos verificar se ele é bom. Para isso, vamos comparar os valores gerados por ele, com os valores reais (os armazenados na base de teste)

Km rodados	Valor cobrado
2	5
3	7
4	9
6	13
4	9
40	81
5	11

2

Modelo  $y = 2x + 1$



3

Valores gerados pelo modelo:

Para  $X = 40$ , temos:  $y = (2 * 40) + 1 = 81$

Para  $X = 5$ , temos:  $y = (2 * 5) + 1 = 11$

Km rodados	Valor cobrado
2	5
3	7
4	9
6	13
4	9
40	81
5	11

**1**  
Base com valores reais

Add a little bit of body text

Valor a ser predito	Real	Predito	Diferença
40	81	81	0
		11	0
Diferença total		0	

**3**



Para  $X = 40 \rightarrow 81$

Para  $X = 5 \rightarrow 11$

**2**  
Use a função score para ver como o algoritmo foi avaliado. Quando mais próximo de 1, melhor.

`reg.score(X_test,y_test)`



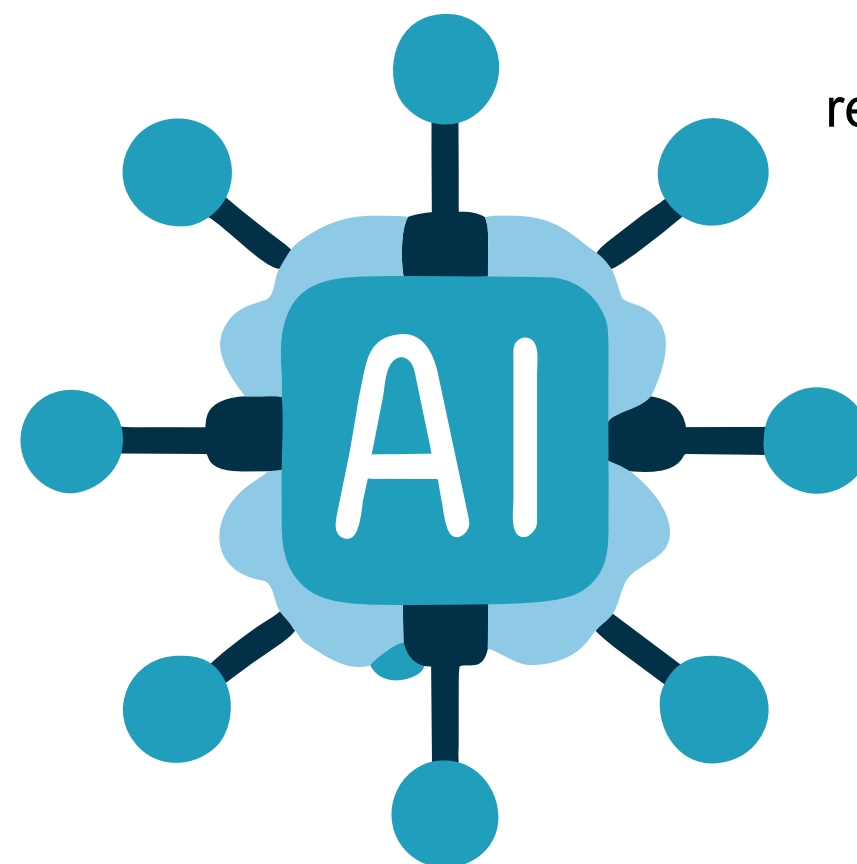
Km rodados	Valor cobrado
2	5
3	7
4	9
6	13
4	9
40	81
5	11

Dados



MMQ

Algoritmos



`reg.fit(X_train, y_train)`

$$y = 2x + 1$$

Modelo

