

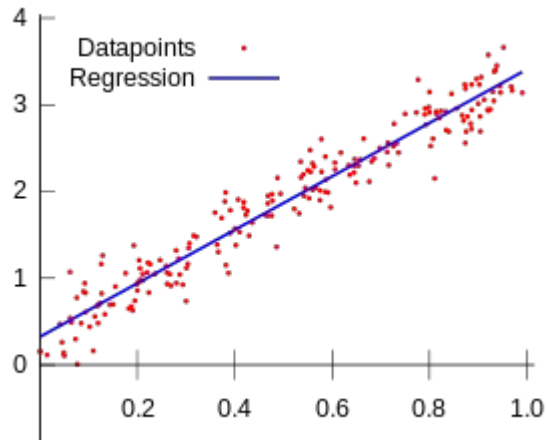


# Modelos de Regressão Linear

# Regressão Linear

A análise de regressão estuda a relação entre uma variável dependente (saída do modelo) e variáveis independentes (entradas do modelo)

- 1(uma) entrada - Regressão Linear Simples
- 2 ou mais entradas - Regressão Linear Múltipla





## Regressão linear - sklearn library

- Importar a biblioteca Linear Regressão
- Determinar as variáveis X's (independente) e Y (dependente) para criação do modelo
- Avaliar ao modelo quanto ao coeficiente de determinação ( $R^2$ ) e p-value

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

```
model = LinearRegression() - Instância o objeto
```

```
model.fit(X,Y) - cria o modelo de regressão linear
```

```
model.score(X, Y) - avalia o modelo quanto ao  $R^2$ 
```

```
** Para regressão múltipla X = [X1,X2,X3, ... , Xn] - n features
```



# Regressão polinomial

A relação entre a variável independente  $x$  (entrada) e a variável dependente  $y$  é modelada como uma  $n$ th função polinomial em  $x$ .

```
from sklearn.preprocessing import PolynomialFeatures
```

```
** Transformar X em X_T -> polinomial
```

```
poly = PolynomialFeatures(degree=3)  
X_T = poly.fit_transform(X) **
```

Usar o método de regressão Linear:  
`lr.fit(X_T,Y)`

```
** X é transformado em features X1, X2, Xn onde X1=X ; X2 = X^2 e Xn = X^n
```

```
** De modo que podemos usar a Regressão Linear para conformar funções não-lineares
```

**Quadratic - 2nd order**

$$\hat{Y} = a + b_1X + b_2X^2$$

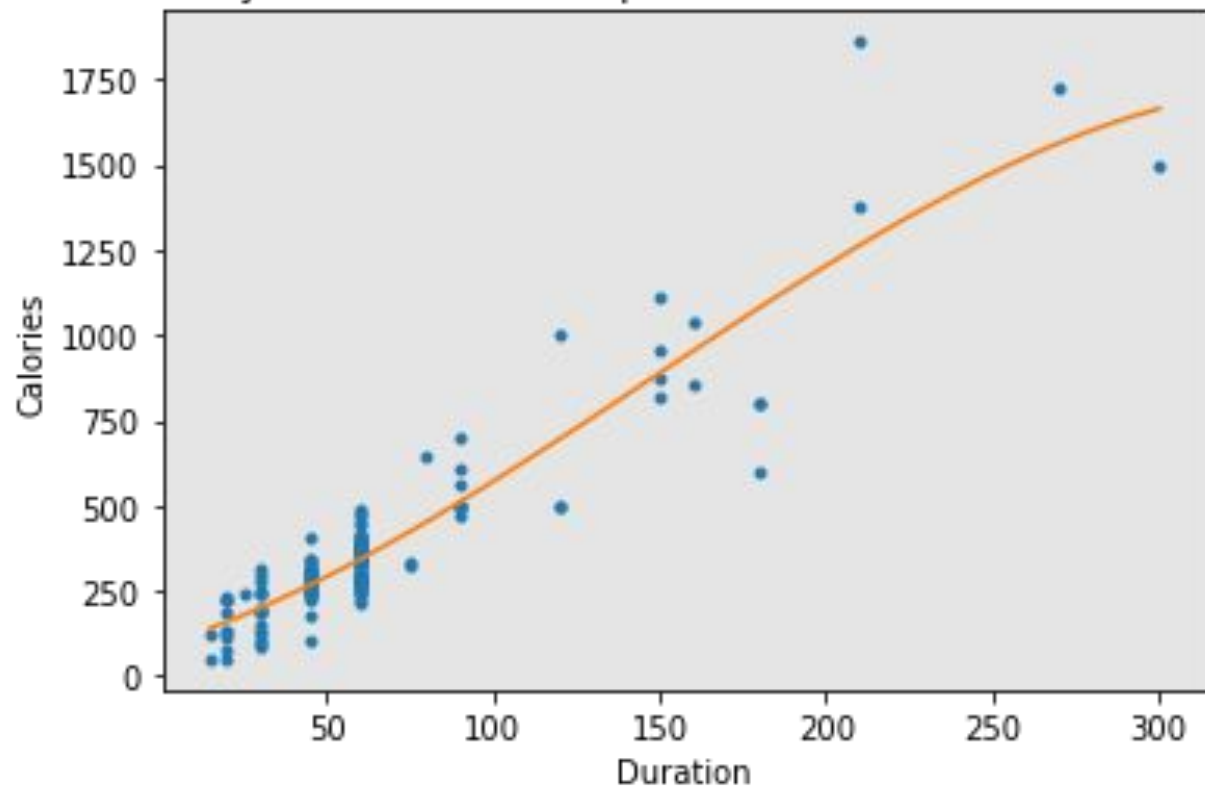
**Cubic - 3rd order**

$$\hat{Y} = a + b_1X + b_2X^2 + b_3X^3$$

**Higher order:**

$$Y = a + b_1X + b_2X^2 + b_3X^3 \dots$$

Polynomial Fit with Matplotlib for Calories ~ Duration

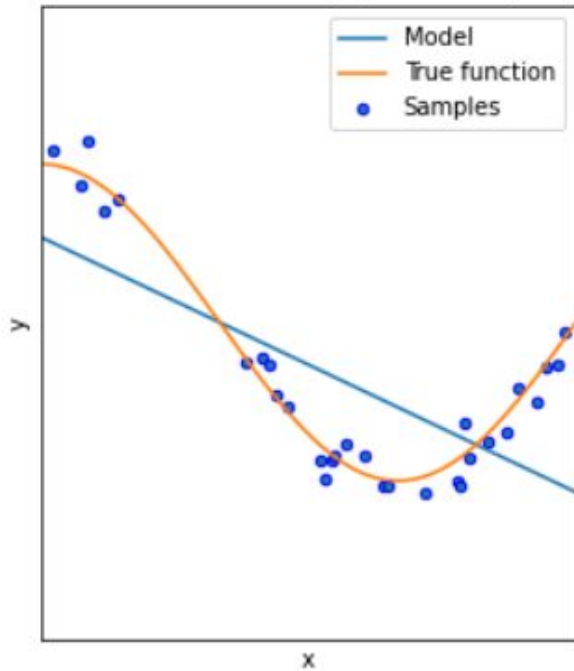




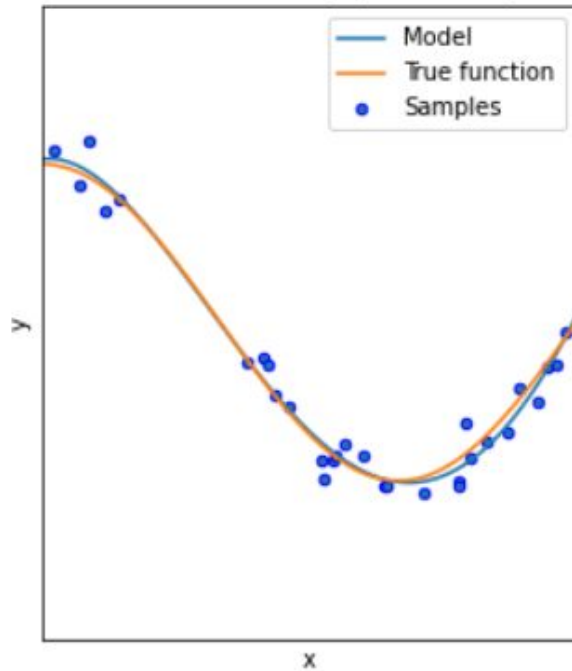
## Construção de Modelos de ML - Regressão

- 1) Dividir o conjunto de dados em treino e teste
- 2) O conjunto de treino é usado para construir o modelo
- 3) O conjunto de teste é usado para avaliar a capacidade do modelo de generalizar. Isto é, se o conjunto de treino representar, estatisticamente, o universo dos dados, o modelo será corretamente aplicado ao conjunto de testes.
- 4) Avaliar o modelo observando o coeficiente de determinação ( $R^2$ ) de ambos os conjuntos, treino e teste. Espera-se que haja, se existir, uma pequena diferença entre eles. Uma diferença muito grande pode ser indicativo de "Overfitting"
- 5) Overfitting - significa que o modelo "aprendeu" muito bem o conjunto de treino, mas não foi capaz de generalizar. Isto é, não tem o mesmo resultado sobre um conjunto de dados "novos" (conjunto de teste)

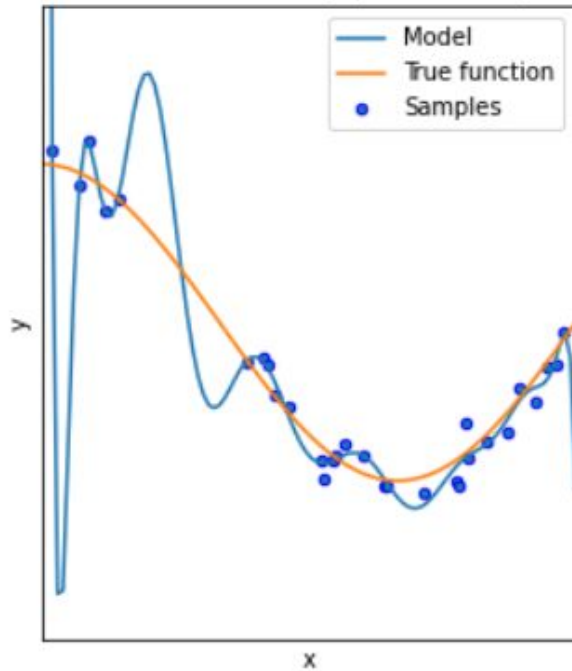
Degree 1  
MSE =  $4.08\text{e-}01$ ( $\pm 4.25\text{e-}01$ )



Degree 4  
MSE =  $4.32\text{e-}02$ ( $\pm 7.08\text{e-}02$ )



Degree 15  
MSE =  $1.81\text{e+}08$ ( $\pm 5.42\text{e+}08$ )





# Métodos para evitar Overfitting

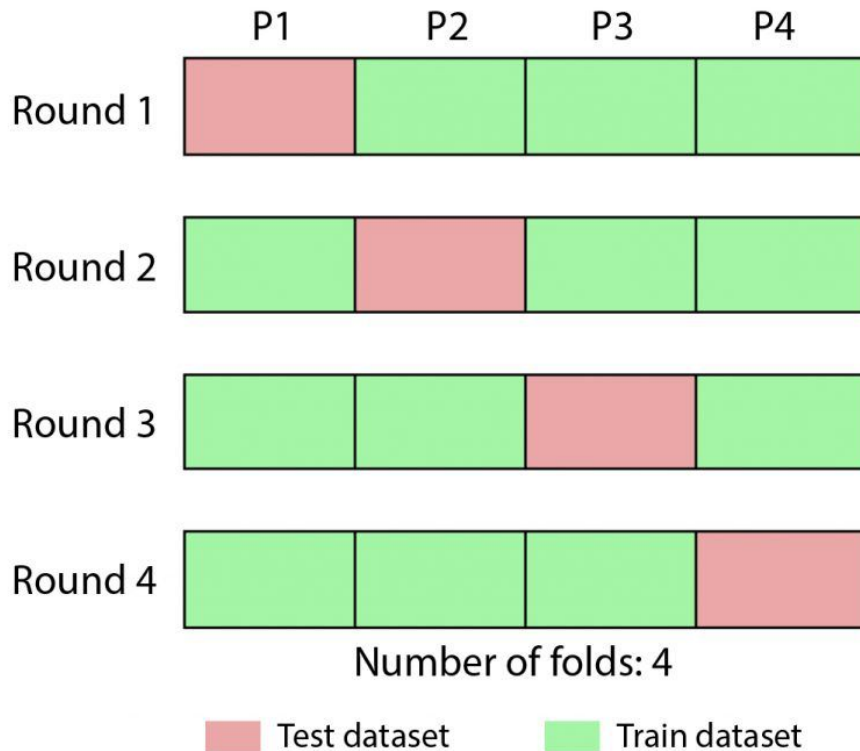
Sobre o Tamanho do conjunto de dados

- Treino/Test Split
- K-fold Cross-Validation
- Leave-one-out cross-validation
- Etc ..





# Cross-Validation





# Métodos para evitar Overfitting

Redução do número de features (X's)

- Principal Component Analysis
- Low Variance Filter
- High Correlation Filter
- Etc..

Regularização

- Lasso
- Ridge Regression
- Dropout



## Regularização L1

$$C = C_0 + \frac{\lambda}{n} \sum_w |w|$$



## Regularização L2

$$C = C_0 + \frac{\lambda}{2n} \sum_w w^2$$



## Ridge - Regressão

- É um método de regressão que restringe os valores dos coeficientes de modo que se evitem valores muito grandes destes coeficientes. Ocorre uma penalização durante o processo de regressão.
- Como consequência ajuda a reduzir a complexidade do modelo e o problema de multicolinearidade.
- É um método que tende a evitar o overfitting