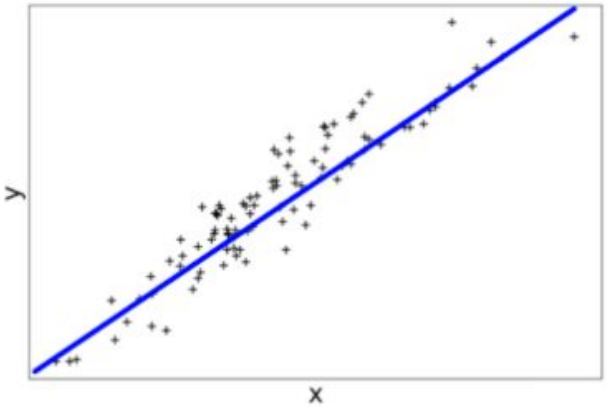
Premier modèle IA

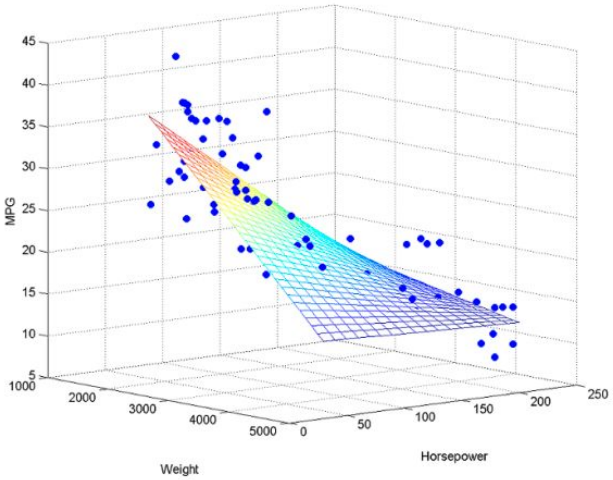
Pereg Hergoualc’h - Paul Sabia

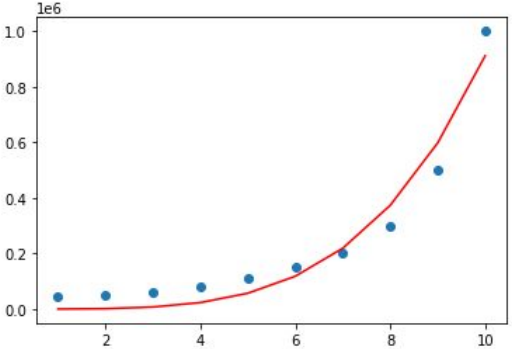
Sommaire

1. Rappel régression linéaire simple multiple et polynomiale
2. Les diverses fonctions
3. Résultat et Évaluation des modèles (Sans et Avec Scikit Learn)
4. Conclusion

Rappel régression linéaire simple, multiple et polynomiale


y
x


MPG
45
40
35
30
25
20
15
10
5
1000
2000
3000
4000
5000
Weight
0
50
100
150
200
250
Horsepower


1e6
1.0
0.8
0.6
0.4
0.2
0.0
2
4
6
8
10

Les diverses fonctions

Model

$$f(x) = a\,x + b$$

$$F = X \cdot \Theta$$

$$\underset{m \times 1}{\begin{bmatrix} f(x^{(1)}) \\ f(x^{(2)}) \\ \vdots \\ f(x^{(m)}) \end{bmatrix}} = \underset{m \times 2}{\begin{bmatrix} x^{(1)} & 1 \\ \vdots & \vdots \\ x^{(m)} & 1 \end{bmatrix}} \cdot \underset{2 \times 1}{\begin{bmatrix} a \\ b \end{bmatrix}}$$

```python
def model(X, theta):
    return X.dot(theta)
```

Fonction coût

$$J(\theta) = \frac{1}{2m} \sum (X\theta - Y)^2$$

```
def fonction_cout(X, y, theta):
    m = len(y)
    return 1/(2*m) * np.sum((model(X, theta) - y)**2)
```

Gradient

$$\frac{\partial J(\theta)}{\partial \theta} = \frac{1}{m} X^T (X\theta - Y)$$

```
def gradient(X,y,theta):
    m = len(y)
    return 1/m * X.T.dot(model(X, theta) - y)
```

Descente de Gradient

$$\theta = \theta - \alpha \frac{\partial J}{\partial \theta}$$

```python
def descente_gradient(X,y,theta,alpha,n_iterations):
    cost = np.zeros(n_iterations)
    for i in range(n_iterations):
        theta = theta - alpha * gradient(X, y, theta)
        cost[i] = fonction_cout(X, y, theta)
    return theta, cost
```

Coefficient de détermination

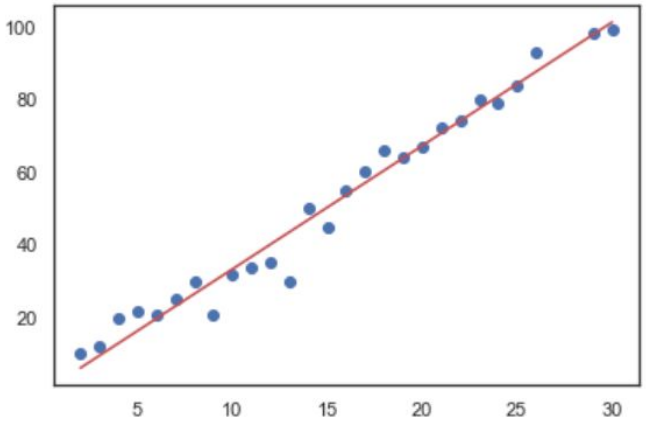
$$R^2 = 1 - \frac{\sum(y - f(x))^2}{\sum(y - \bar{y})^2}$$

```python
def coef_determination(X, y, theta):
    return 1 - ((y - model(X, theta))**2).sum() / ((y - y.mean())**2).sum()
```

Résultats des modèles

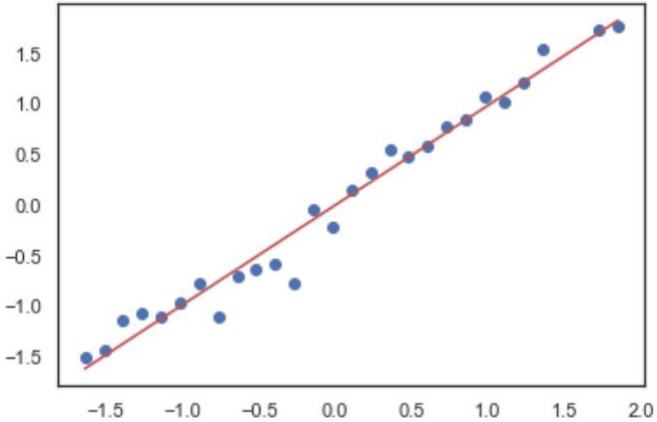
Régression linéaire

Sans Scikit-Learn



Score: 0.9729

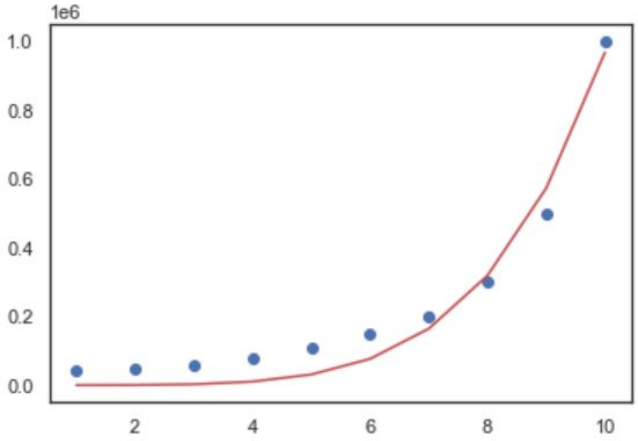
Avec Scikit-Learn



Score: 0.9733

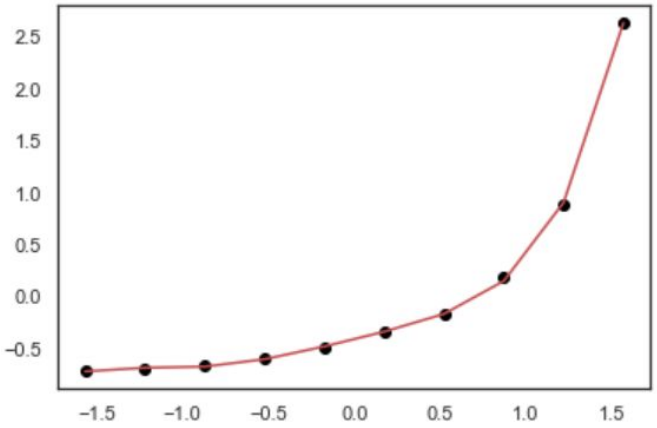
Régression polynomiale

Sans Scikit-Learn



Score: 0.9598

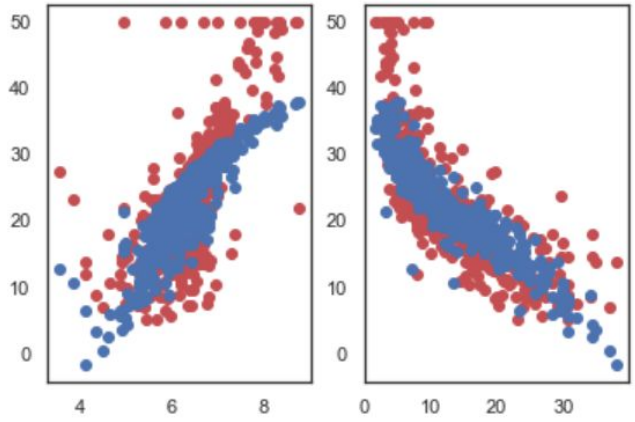
Avec Scikit-Learn



Score: 0.9997

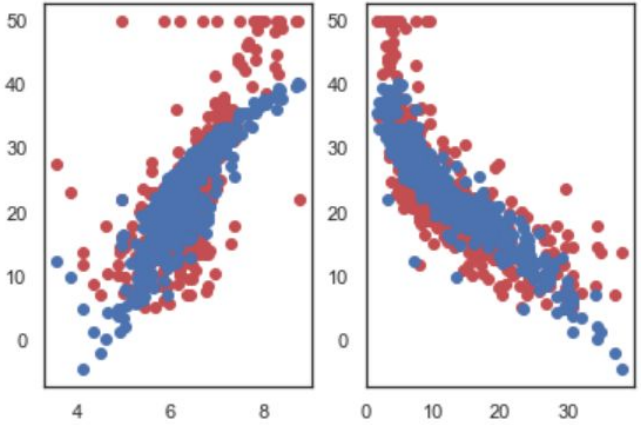
Régression multiple

Sans Scikit-Learn



Score: 0.6282

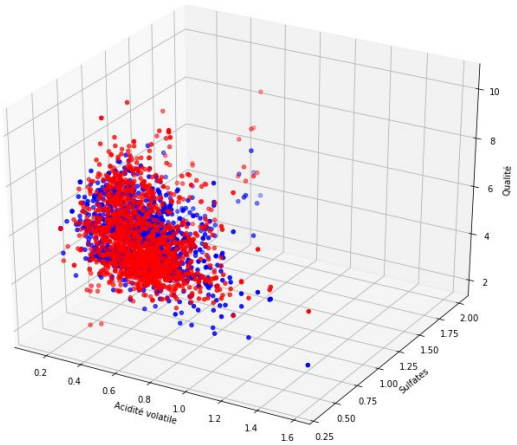
Avec Scikit-Learn



Score: 0.6385

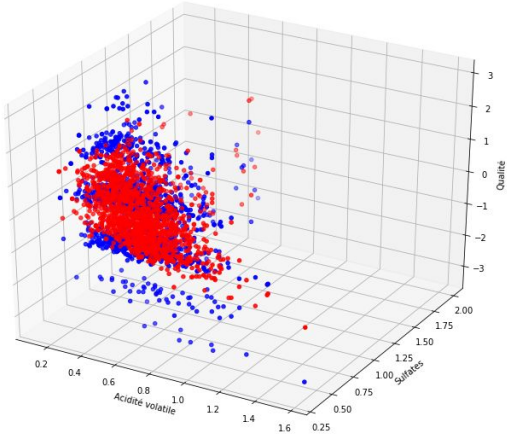
Régression multiple (vins)

Sans Scikit-Learn



Score: -0.65(?)

Avec Scikit-Learn



Score: 0.36

Conclusion