Apprentissage automatique supervisé par régression linéaire

Présenté par : SMAHI AMINE

Sommaire

- ✘ Qu'est-ce qu'un apprentissage automatique supervisé
- ✘ Type d'apprentissage machine supervisé
- ✘ Qu'est-ce que la régression et ces type
- ✘ Comprendre la régression linéaire
- ✘ Exercice pratique sur la régression linéaire avec Python

Qu'est-ce que l'apprentissage supervisé?

En apprentissage supervisé, nous recevons un ensemble de données étiquetées et le résultat souhaité est déjà connu,

L'apprentissage supervisé consiste en des variables d'entrée (x) et une variable de sortie (Y) . On utilise un algorithme pour apprendre la fonction de mappage de l'entrée à la sortie.

n

$$Y = f(X)$$

Type d'apprentissage machine supervisé

La manière la plus fondamentale de catégoriser une méthodologie d'apprentissage supervisé est basée sur le type d'énoncé du problème qu'elle tente de résoudre.

- Régression
- Classification

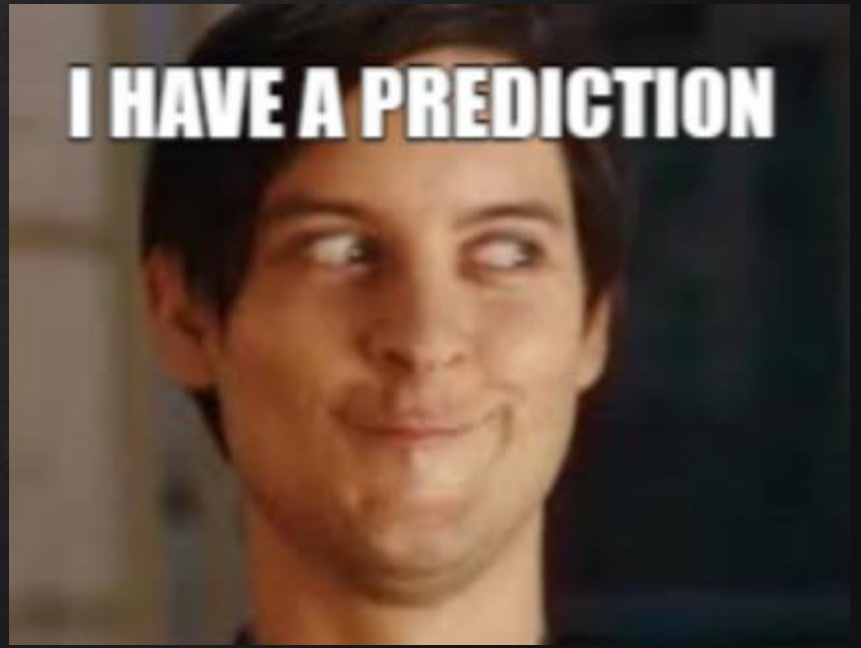
La manière la plus fondamentale de catégoriser une méthodologie d'apprentissage supervisé est basée sur le type d'énoncé du problème qu'elle tente de résoudre.

- Régression
- Classification

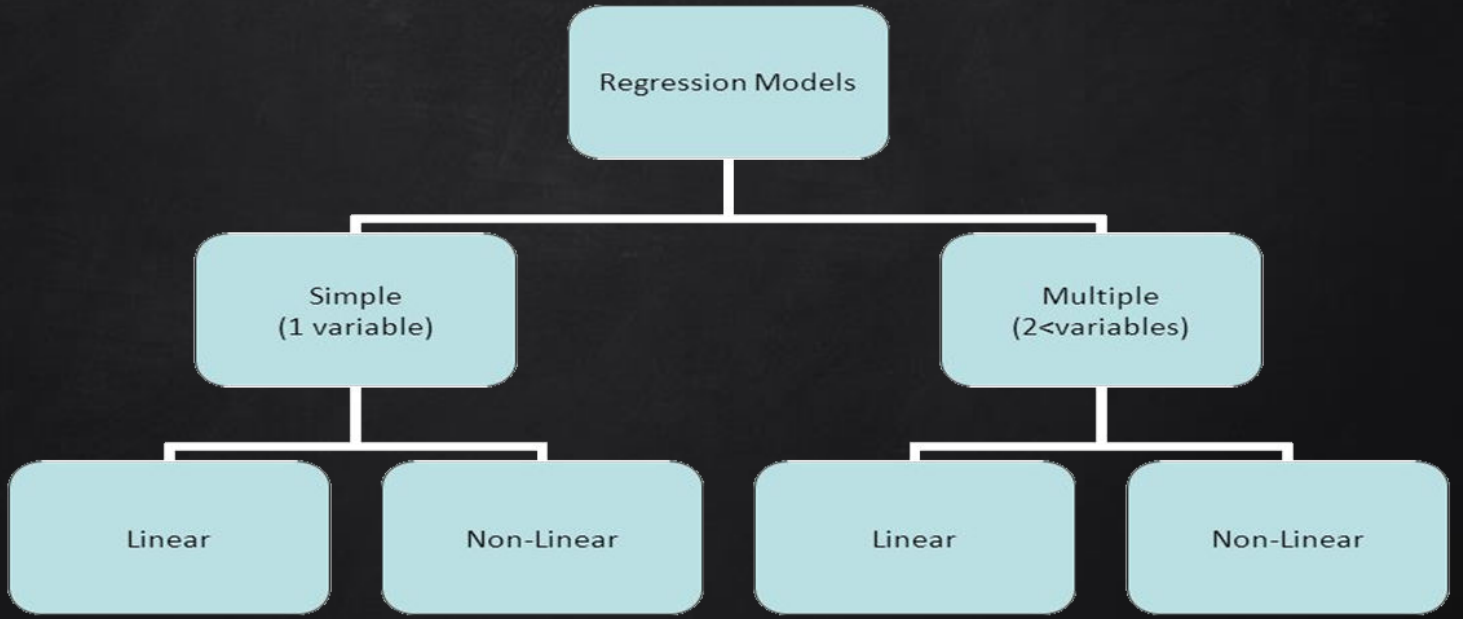
Qu'est-ce que la régression et ces types

Les problèmes de régression sont les problèmes pour lesquels nous essayons de faire une prédiction à une échelle continue.

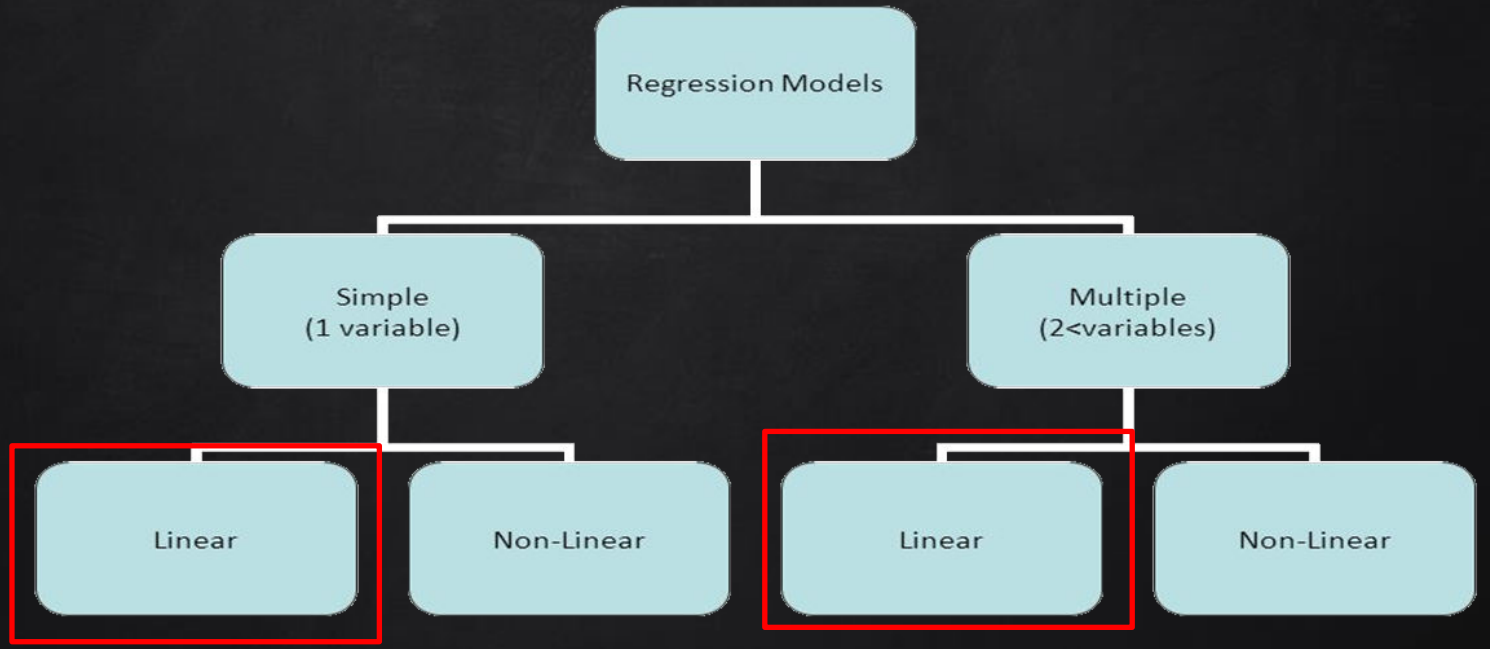
Des exemples pourraient être la prédiction du prix des actions d'une entreprise ou la prédiction de la température demain sur la base de données historiques.



Les Types de régressions



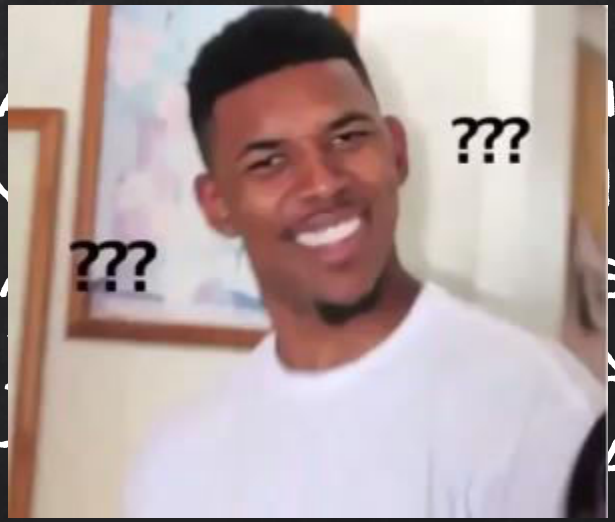
Les Types de régressions



Qu'est-ce que la régression linéaire?

Il s'agit d'une approche statistique consistant à modéliser une variable dépendante et une ou plusieurs variables explicatives (ou variables indépendantes) afin d'obtenir la droite linéaire la mieux ajustée (équation linéaire)

Il s'agit d'une a ique consistant à modéliser un dante et une ou plusieurs vari s (ou variables indépendantes) a droite linéaire la mieux ajust aire)



Régression linéaire simple

$$Y = A X + B$$

X = variables explicatives,
B = ordonnée à l'origine (constant),
A = coefficients de pente pour la variable explicative,

Régression linéaire simple

$$Y = A X + B$$

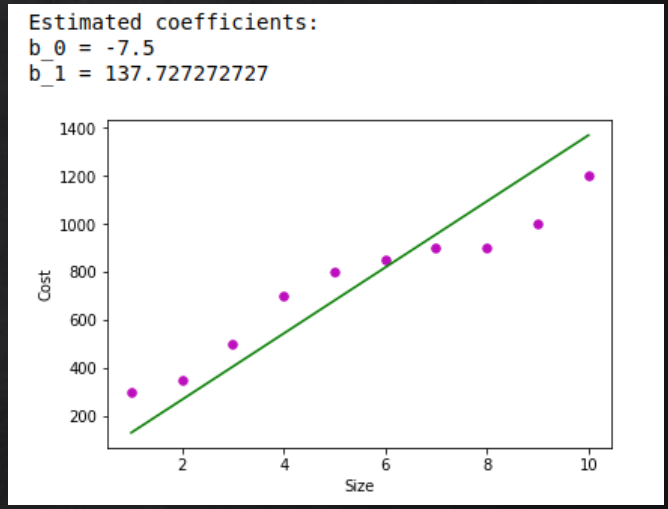
la regression linéaire multiple

$$y = B + A1\,x1 + A2\,x2 + \dots$$

X = variables explicatives,
B = ordonnée à l'origine (constant),
A = coefficients de pente pour la variable explicative,

Exercice pratique sur la régression linéaire avec Python

X	1	2	3	4	5	6	7	8	9	10
Y	300	350	500	700	800	850	900	900	1000	1200



X – Taille de la maison de 1K pieds carrés à 10K pieds carrés.
Y – Coût de la maison de 300K à 1200K.

ETAPE 1

```
import numpy as np
import matplotlib.pyplot as plt

def main():
    # Datasets which we create
    x = np.array([ 1,   2,   3,   4,   5,   6,   7,   8,    9,   10])
    y = np.array([300, 350, 500, 700, 800, 850, 900, 900, 1000, 1200])

    # estimating coefficients
    b = estimate_coefficients(x, y)
    print("Estimated coefficients:\nb_0 = {} \nb_1 = {}".format(b[0], b[1]))

    # plotting regression line
    plot_regression_line(x, y, b)

if __name__ == "__main__":
    main()
```

ETAPE 2

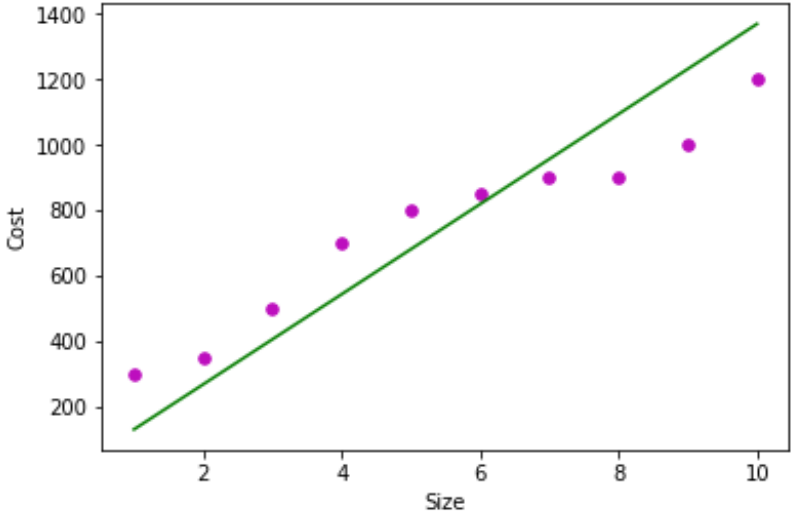
```
def plot_regression_line(x, y, b):
    # plotting the points as per dataset on a graph
    plt.scatter(x, y, color = "m",marker = "o", s = 30)

    # predicted response vector
    y_pred = b[0] + b[1]*x

    # plotting the regression line
    plt.plot(x, y_pred, color = "g")

    # putting labels for x and y axis
    plt.xlabel('Size')
    plt.ylabel('Cost')

    # function to show plotted graph
    plt.show()
```

ETAPE 3

```python
def estimate_coefficients(x, y):
    # size of the dataset OR number of observations/points
    n = np.size(x)

    # mean of x and y
    # Since we are using numpy just calling mean on numpy is sufficient
    mean_x, mean_y = np.mean(x), np.mean(y)

    # calculating cross-deviation and deviation about x
    SS_xy = np.sum(y*x - n*mean_y*mean_x)
    SS_xx = np.sum(x*x - n*mean_x*mean_x)

    # calculating regression coefficients
    b_1 = SS_xy / SS_xx
    b_0 = mean_y - b_1*mean_x

    return(b_0, b_1)
```

Résultat

```
Estimated coefficients:
b_0 = -7.5
b_1 = 137.727272727
```



Limites

- ✘ La régression linéaire est limitée aux relations linéaires
- ✘ La régression linéaire ne regarde que la moyenne
- ✘ Les données doivent être indépendantes

MERCI!