

Data Science et Machine Learning

TP2 sur le modèle de la régression linéaire A réaliser en Python avec la bibliothèque Scikit-learn

Nous expliquons le coût de fonctionnement d'une entreprise avec les quantités fabriquées. Nous disposons pour cela des données de 12 mois déjà fournies dans le fichier Production.xlsx.

1. Réalisez une régression linéaire simple avec la fonction `LinearRegression` du module `linear_model` de la bibliothèque Scikit-learn pour expliquer le coût de fonctionnement avec la quantité fabriquée du premier produit. Affichez le coefficient de détermination de ce modèle qui peut être obtenu avec la méthode `score`. Affichez les deux paramètres de la régression linéaire simple qui se trouvent dans l'attribut `intercept_` et l'attribut `coef_`.

Indication :

Les deux vecteurs `x` et `y` sont d'une seule dimension mais `x` doit être redimensionné pour être accepté par la fonction `LinearRegression` :

```
import numpy as np
from sklearn.linear_model import LinearRegression
import openpyxl
file = 'Production.xlsx'
wb = openpyxl.load_workbook(file)
ws = wb["données"]
n = 12
y = [ws.cell(2+i,2).value for i in range(n)]
x = [ws.cell(2+i,3).value for i in range(n)]
x = np.array(x).reshape(-1, 1)
y = np.array(y)
reg = LinearRegression().fit(x, y)
R2 = reg.score(x, y)
print('R2:', R2)
print('La constante :', reg.intercept_)
print('Le coefficient :', reg.coef_)
```

2. Ecrivez un nouveau programme Python qui utilise la bibliothèque Scikit-learn pour trouver les paramètres d'une régression linéaire multiple qui explique le coût de fonctionnement avec la quantité fabriquée des deux premiers produits.

Indication :

Utilisez le même programme que la régression linéaire simple où `X` devient une matrice avec deux colonnes et où il n'y a plus besoin de redimensionner `X`.