

TP3 : REGRESSION

Les modèles de Régression sont utilisés pour prédire une valeur réelle continue, comme par exemple le salaire. Si votre variable indépendante est le temps, alors votre modèle prédit des valeurs futures. Sinon, votre modèle prédit des valeurs présentes, mais inconnues.

Dans cette partie nous allons former notre intuition et faire l'implémentation des modèles de régression suivants:

- Régression Linéaire Simple
- Régression Linéaire Multiple
- Régression Polynomiale

Données : Salary_Data.csv

A. Régression Linéaire Simple

DATA PREPROCESSING

E 1 : Importer les librairies

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

E 3 : Importer le dataset, extraire la matrice des variables indépendantes et le vecteur des variables dépendantes.

```
dataset = pd.read_csv('Salary_Data.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

E 4 : Diviser le dataset entre le Training set et le Test set

E 4-1 : importer la librairie train_test_split

```
from sklearn.model_selection import train_test_split
```

E 4-2 : construire le training (2/3) et le test (1/3) set

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1.0/3,
random_state = 0)
```

NB : il n'y a pas de feature scaling sur les modèles de régression linéaires simple

CONSTRUCTION DU MODELE DE REGRESSION LINEAIRE SIMPLE

a. # Construction du modèle

importer la classe LinearRegression

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

instancier la classe

```
regressor = LinearRegression()
```

Lier cet objet à notre training set

La méthode des moindres carrés ordinaires va s'exécuter pour trouver les coefficients optimaux de sorte que la droite de régression se rapproche le plus de l'ensemble des points en terme de somme des distances au carré entre les valeurs observées et les valeurs prédites.

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

Faire de nouvelles prédictions valeur predites par le modele de Regression Lineaire Simple (RLS)

```
y_pred = regressor.predict(X_test)
```

comparer valeurs predites et valeurs reels

.....

prediction sur une nouvelle valeur

```
regressor.predict(15)
```

Visualiser les results du predicteur sur le Training set

```
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Training set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```

Visualiser les résultats du prédicteur sur le Test set

```
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Test set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```

A. Régression Linéaire multiple

