

**Série TP n°=2**

**Exercice 1 : Régression**

Etant donnée la transformation matricielle du problème de régression linéaire comme suit :

| ID                  | Représentation matricielle   |
|---------------------|--|
| Model               | $F = X * \theta$   |
| Fonction cout       | $J(\theta) = \frac{1}{2} \sum (X\theta - Y)^2$                           |
| Gradient            | $\frac{\delta J(\theta)}{\delta \theta} = \frac{1}{m} X^T (X\theta - Y)$ |
| Décence de gradient | $\theta = \theta - X \frac{\delta J(\theta)}{\delta \theta}$             |

1. Ecrire en python le programme de la régression linéaire en utilisant la représentation matricielle ci-dessus

**Exercice 2 : SVM**

**Partie 1 :**

1. Charger les données Iris
2. Afficher les différents attributs
3. Diviser les données comme suit : 70% jeu d'entraînement et 30% jeu de test
4. Entraîner le modèle en utilisant une SVM linéaire avec  $c=0,05$  et  $\text{gamma}='auto'$
5. Prédire le jeu de données test
6. Afficher la matrice de confusion
7. En utilisant méthode gridsearch de Scikit-learn, trouver les valeurs de C et gamma qui donnent les meilleures performances

**Partie 2 :**

**Essayer la solution précédente avec d'autre kernels comme :**

1. Classifier=SVC( $c=0.006$ , kernel='rbf', gamma='scale')
2. Classifier=SVC( $c=1$ , kernel='poly', gamma='auto', degree=7)

**Partie 3 :**

Utiliser le jeu de données *breast\_cancer* de Scikit-Learn et tenter de créer une SVM qui permet de différencier tumeurs malignes des tumeurs bénignes