

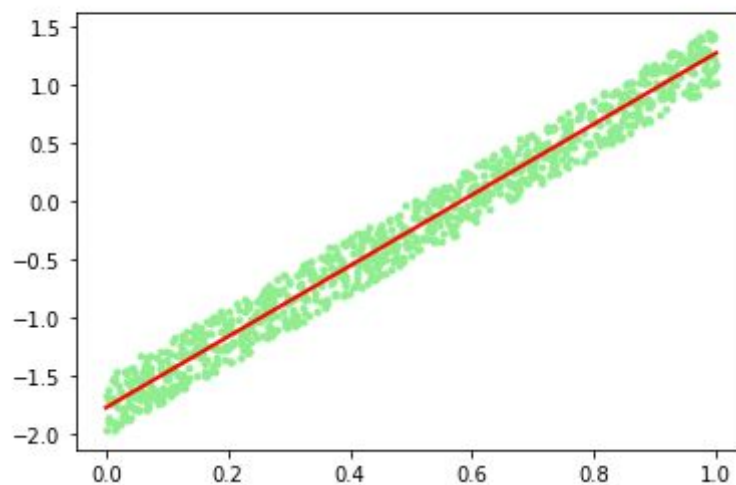
Rapport TME 1 : regression

1)

Les résultats obtenus suite à l'implémentation de l'algorithme de régression linéaire sont les mêmes qu'obtenus avec `linregress` de la bibliothèque `np`.

Résultats :

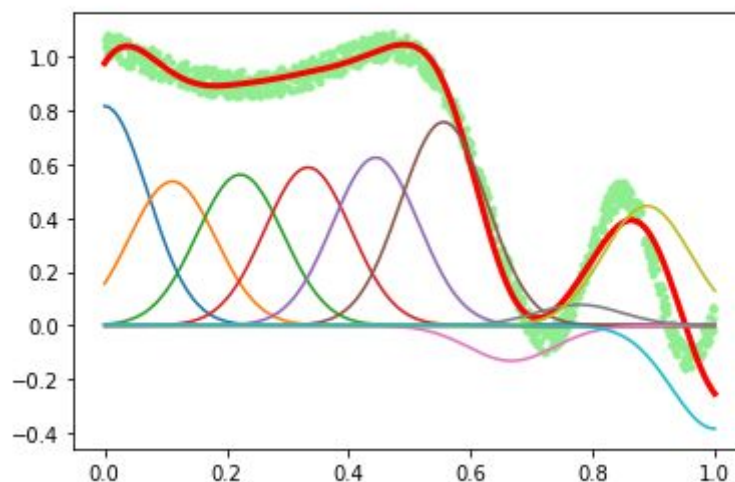
```
theta: [ 3.04090111 -1.77072827]  
time: 0.001531517000003646
```



2)

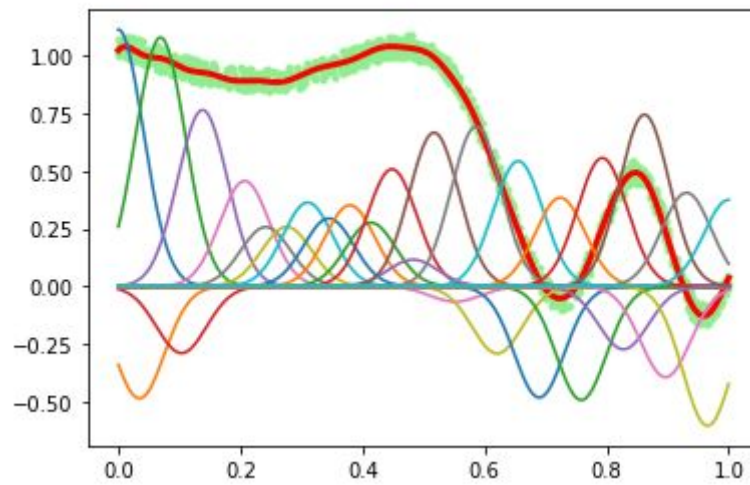
Après plusieurs essais, nous avons trouvé qu'un `nb_features` de 30 fournit de bons résultats. Avec un `nb_feature` de 10, la fonction est assez mal approximée au endroits avec fort changement de pente.

```
2483.1129223563453  
time: 0.002710139999997807
```



Ci dessus avec nb_feature = 10, en dessous avec nb_feature = 30

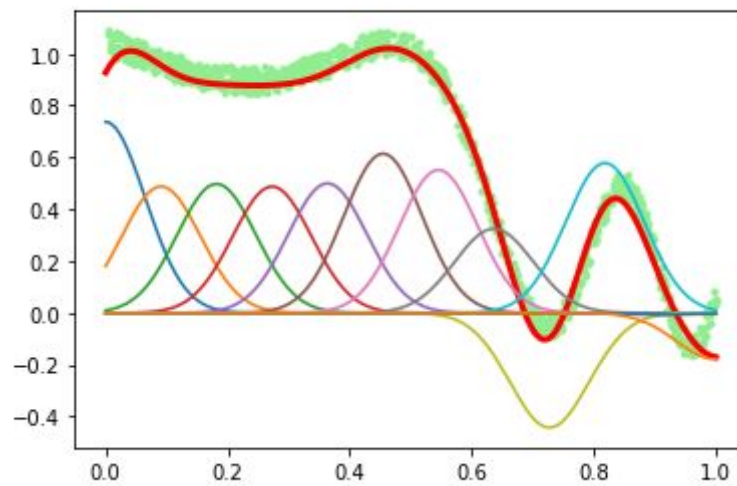
time: 0.007272617999994679



3)

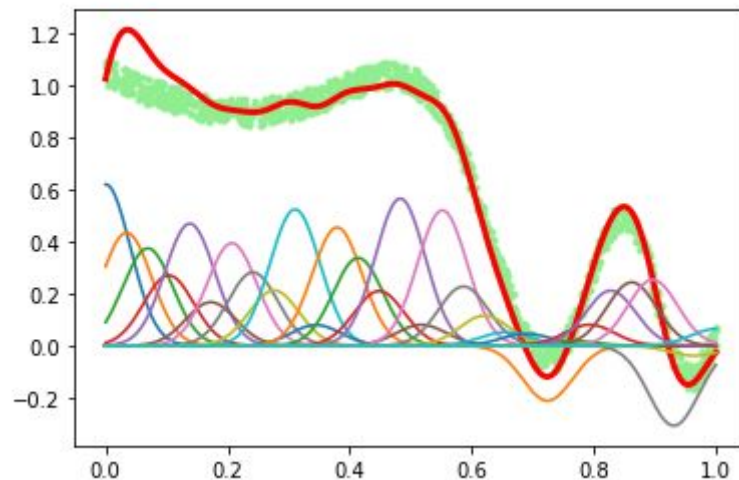
A un nb_feature de 10, l'apprentissage n'est pas complet, à partir d'un nb_feature de 15, il commence à y avoir du sur apprentissage. Nous avons trouvé qu'un nb_feature de 12 rendait les meilleurs résultats à l'exception d'une erreur vers 0..

time: 0.022823899999999675



ci dessus avec nb_feature = 10, ci dessous avec nb_feature = 12

time: 0.023498845999999985



4)

Nous ne sommes pas parvenus à faire fonctionner la régression locale pondérée, nous avons eu un problème de dimensions lors du calcul de B, y n'ayant pas les dimensions adéquates pour le produit avec le reste du produit matriciel.

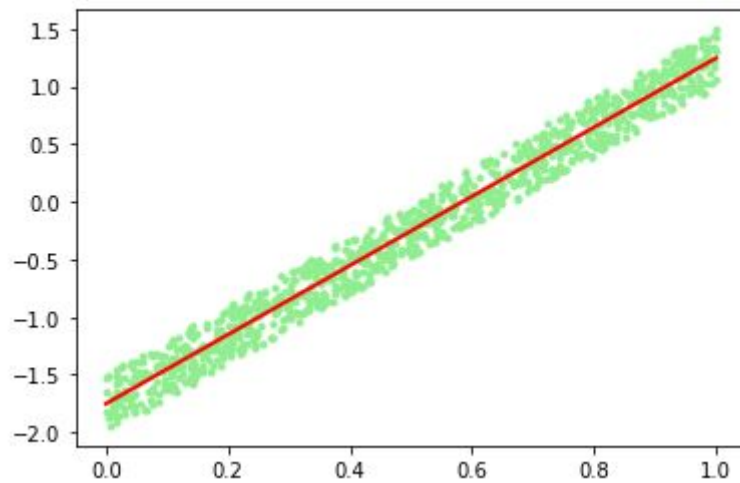
Voici notre code :

```
def train_lwls(self, x_data, y_data):  
    X = np.array(x_data)  
    Y = np.array(y_data)  
    w = weight(X)  
    phi = self.phi_output(X)  
  
    for i in range(self.nb_features):  
        print(phi[i].shape)  
        print(w.shape)  
        print(Y.shape)  
  
        print(np.dot(phi[i],w.T))  
  
        self.theta[0,i] = np.sum(np.dot(np.dot(phi[i],w.T),w))  
        self.theta[1,i] = np.sum(np.dot(np.dot(phi[i],w.T),Y))
```

5) Voir code.

6) On peut voir que moins il y a de points et plus le coefficient est haut plus la régularisation est élevée.

theta regularized: [2.99651994 -1.75148184]



7) Avec une régularisation de 0,1 on obtient un theta de 3,0101 et -1.7578 et un theta régularisé de 3,0054 et -1,7553 et avec un coefficient de 0,9 pour un theta original de 3.0216 et -1,7595 on a un theta régularisé de 2,9801 et -1,7378.