

Intelligence Artificielle

1. Régression linéaire

On donne (sur moodle) le corps du programme qui traite et affiche les données du TD n°1.

a) Vous devez ajouter le calcul de l'estimateur des moindres carrés dans le cas d'une régression linéaire.

Pour cela on utilise la formule suivante :

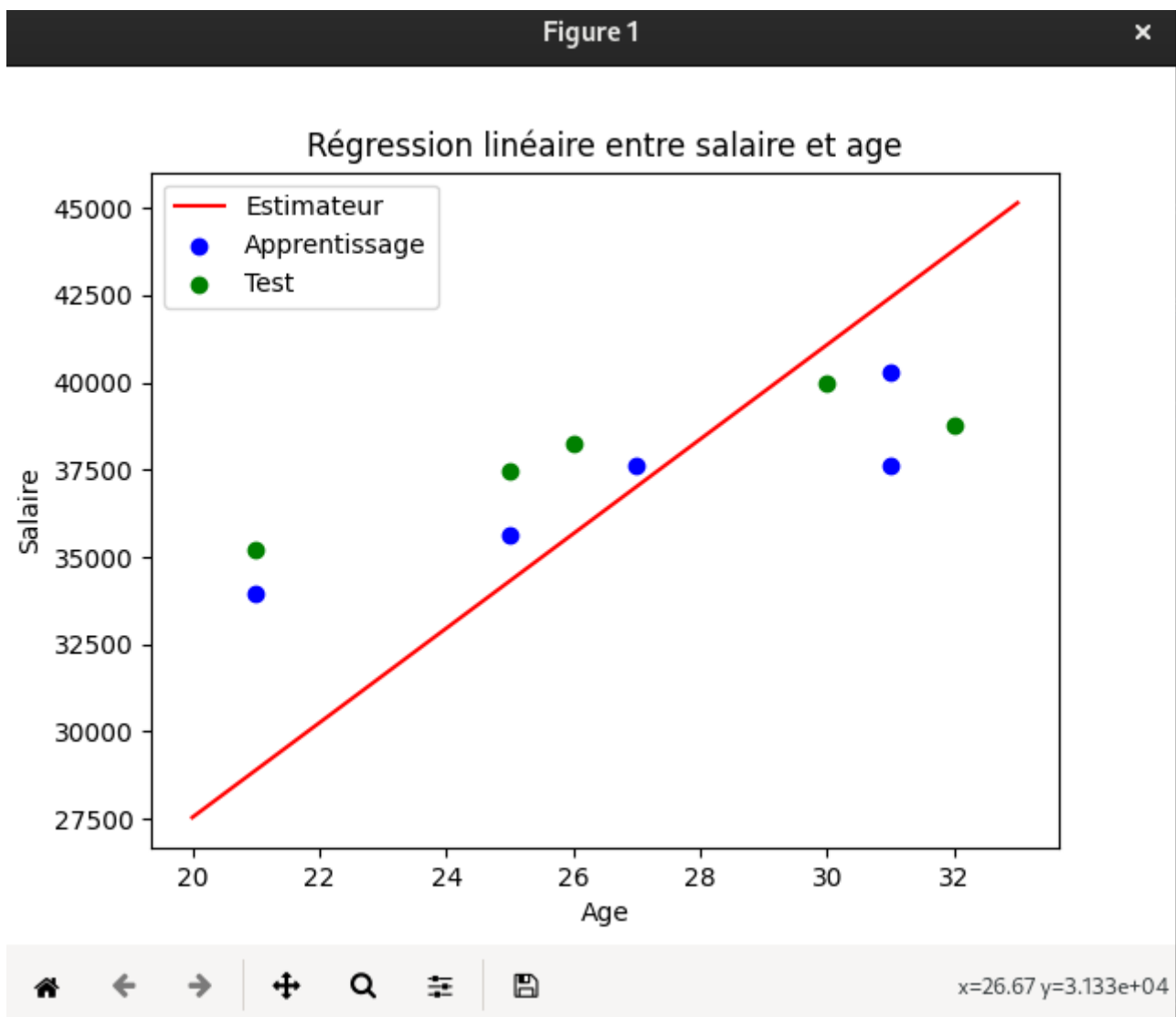
$$\beta = (X^T X)^{-1} X^T Y$$

ou X est le vecteur correspondant à l'âge et Y le vecteur des salaires.

On utilise donc la commande de numpy `np.linalg.inv` pour calculer l'inverse d'une matrice.

ainsi qu'un reshape pour les vecteurs X et Y.

Pour la range de 42 :

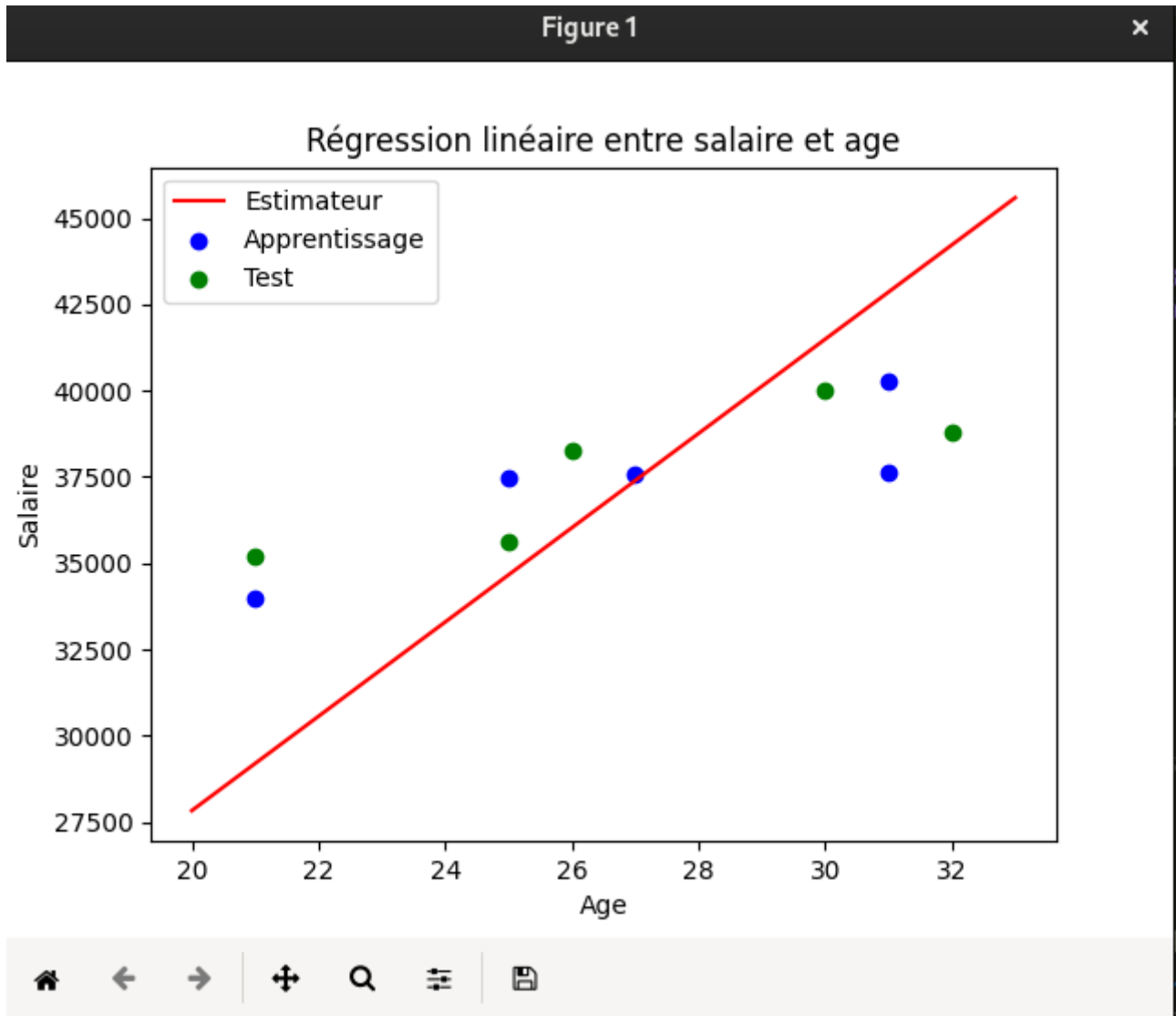


Erreur quadratique moyenne d'apprentissage : 3329.82407780037316

Erreur quadratique moyenne de test 4065.720768997446

Coefficient de détermination (score R2) : -5.461791196554768

Pour la range de 60 :



Erreur quadratique moyenne d'apprentissage 3596.418588284633 Erreur quadratique moyenne de test 3841.8521234280156 Coefficient de détermination (score R2) : -3.2655636043671192

b) Mesurer et commenter l'efficacité de votre apprentissage pour plusieurs mélanges du dataset. Il n'y a pas assez de points d'apprentissage pour que la mesure de l'efficacité soit vraiment intéressante.

c) Modifier le programme pour générer les données et vérifier une loi classique en IA : si on a peu de données d'apprentissage l'erreur d'apprentissage est faible et l'efficacité médiocre et vice-versa.

On pourra, par exemple fixer, a_0 et a_1 et générer des couples (age, salaire) en ajoutant une valeur aléatoire au salaire :

`age=rng.integers(low=20, high=40, size=N) salaire = 35000 + 250*age + rng.normal(0, 1000, size=N)`