



ft_linear_regression

Une introduction au machine learning

Résumé: Dans ce projet vous implementerez votre premier algorithme de machine learning.

Table des matières

I	Préambule	2
II	Introduction	3
III	Objectif	4
IV	Instructions Générales	5
V	Ce que vous devez faire	6
VI	Bonuses	7
VII	Peer-evaluation	8

Chapitre I

Préambule

Ce que je pense être la meilleure définition du machine learning :

"On dit d'un programme informatique qu'il apprend d'une expérience E en respectant les règles d'une tâche T et en mesurant les performances via P si ses performances à la tâche T , telles que mesurées par P , s'améliorent avec l'expérience E ."

Tom M. Mitchell

Chapitre II

Introduction

Machine learning est un des champs de l'informatique qui est en pleine expansion. Ça à l'air compliqué et réservé aux mathématiciens au premier abord. Vous avez sûrement entendu parler de réseaux de neurones ou bien k-moyennes et vous ne comprenez pas encore comment ça marche ou comment développer de tels algorithmes.

Ne vous inquiétez pas, on va commencer par les bases, un simple algo de machine learning

Chapitre III

Objectif

Le but de ce projet est de vous faire une introduction au concept de base du machine learning. Pour ce projet vous devrez créer un programme qui prédit le prix d'une voiture en utilisant l'entraînement par [fonction linéaire](#) avec un [algorithme du gradient](#).

Nous travaillerons sur un exemple précis pour ce projet, mais une fois que vous aurez fini vous pourrez l'utiliser sur d'autres jeux de données.

Chapitre IV

Instructions Générales

Dans ce projet vous êtes libres d'utiliser n'importe quel langage de programmation tant qu'il ne font pas tout le travail pour vous.
Par exemple, python et numpy.polyfit sera considéré comme de la triche.



Utilisez un langage ou il est facile de visualiser des données, ça vous aidera à debugguer.

Chapitre V

Ce que vous devez faire

Implémentez un algorithme de regression linéaire sur un seul element, en l'occurrence le kilométrage d'une voiture Pour ce faire vous devez faire 2 programmes :

- Le premier programme sera utilisé pour prédire le prix d'une voiture en fonction de son kilométrage. Quand vous lancerez le programme, celui ci vous demandera le kilométrage et devrait vous donner un prix approximatif de la voiture en utilisant l'hypothèse suivante :

$$prixEstime(kilométrage) = \theta_0 + (\theta_1 * kilométrage)$$

Avant de lancer le programme d'entrainement, theta0 et theta1 auront pour valeur 0.

- Le second programme sera utilisé pour entrainer votre modèle. Il lira le jeu de données et fera une regression linéaire sur ces données.
Une fois la regression linéaire terminée, vous sauvegarderez la valeur de theta0 et theta1 pour pouvoir l'utiliser dans le premier programme.
Vous utiliserez la formule suivante :

$$tmp\theta_0 = ratioDApprentissage * \frac{1}{m} \sum_{i=0}^{m-1} (prixEstime(kilométrage[i]) - prix[i])$$

$$tmp\theta_1 = ratioDApprentissage * \frac{1}{m} \sum_{i=0}^{m-1} (prixEstime(kilométrage[i]) - prix[i]) * kilométrage[i]$$

Je vous laisse devinez ce à quoi m correspond :)

Veuillez noter que le prixEstime est la même chose que dans notre premier programme, mais ici il utilise vos valeurs temporaires afin de calculer theta0 et theta1. Attention a bien mettre a jour theta0 et theta1 en même temps.

Chapitre VI

Bonuses

Voici des bonus qui pourrait être utiles :

- Visualiser les données sur un graph avec leur repartition.
- Afficher la ligne résultant de votre regression linéaire sur ce même graphe et voir si ca marche!
- Un programme qui vérifie la precision de votre algorithme.

... Et n'importe quoi qui pourrait rendre cet exercice encore meilleur.

Chapitre VII

Peer-evaluation

Votre programme ne sera corrigé que par des humains, vous pouvez donc organiser votre code comme bon vous semble.

Voici les points qui seront vérifiés par votre peer-correcteur :

- L'absence de librairie ou autre qui fasse tout le travail à votre place.
- L'utilisation de l'hypothèse donnée plus haut
- L'utilisation de la fonction d'entraînement donnée plus haut.