Ecole National Polytechnique ENP Spécialité : DSIA	2°Année du 2°Cycle Module : Apprentissage Automatique(AA) Année 2021-2022

TP 01 : Regression linaire

L'objectif :

L'objectif de ce TP est de construire notre premier modèle de machine learning

Les trois ingrédients d'un algorithme d'apprentissage supervisé sont :

- l'espace des hypothèses (dans notre cas c'est l'espace des droites)
- la fonction de coût (le coût quadratique)
- l'algorithme d'optimisation qui permet de trouver l'hypothèse optimale au sens de la fonction de coût sur les données (minimisation du risque empirique), donc la descente de gradient.

Partie 01 : Regression simple : $f(x_i) = b_0 + b_1 x_i$

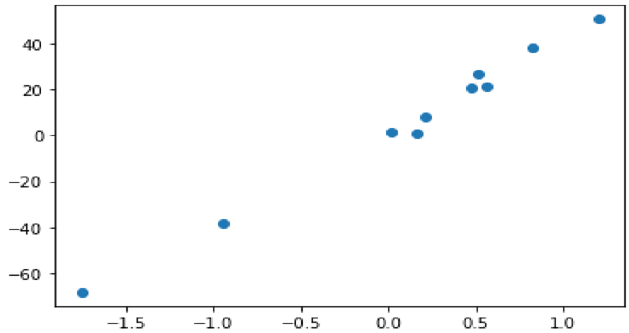
Pour le jeu de données D={(xi,yi)},on peut utiliser le module suivant pour la génération du D

from sklearn.datasets import make_regression

x,y=make_regression(n_samples=10,n_features=1,noise=3)

pour visualuer le nuage du point :

plt.scatter(x,y)



Méthode 01 :

Dans un premier temps nous allons utiliser la méthode MMC pour l'estimation des coefficients :

$$\hat{b}_0 = \bar{y} - \hat{b}_1\bar{x}$$

$$\hat{b}_1 = \frac{cov(x, y)}{var(x)} = r(x,y)\sqrt{\frac{var(y)}{var(x)}}$$

Pour ce faire nous allons utiliser la bibliothèque numpy pour trouver les différentes caractéristiques

Moyenne: $\bar{x}=\frac{1}{n}\sum_{i=1}^{n} xi$ **Variance** : $V(x)=\frac{1}{n}\sum_{i=1}^{n}(xi-\bar{x})^2$ **Ecart type**: $\sigma_x=\sqrt{V(x)}$

Covariance: $Cov(x, y)=\frac{1}{n}\sum_{i=1}^{n}(xi-\bar{x})(y_i-\bar{y})$ **Coefficient de corrélation** : $Corrcoef(x,y)=\frac{Cov(x, y)}{\sigma_x\sigma_y}$

Méthode 02 : La descente de gradient

Dans cette partie nous allons utiliser un algorithme d'optimisation pour trouver les meilleurs paramètres de la droite de régression

Algorithme du gradient (descente de gradient)

1-Initialiser avec $\mathbf{p_0}$(point choisi au hasard)

2-Répéter

$$\mathbf{p}_{t+1}=\mathbf{p}_t-\alpha\times\nabla(\mathbf{p}_t)$$

3-Jusqu'à **convergence** (nombre d'itérations fixé, ou $|\nabla(\mathbf{p}_{t+1})|$ **très petit ≈*null*)**

Démarche :

Les variables :

La variable y : la matrice colonne des étiquettes

La variable X : une matrice des observations (avec une colonne des 1 à la fin)

La variable theta : une matrice colonne des paramètres du model (b1, b0 respectivement)

1/Créer une fonction **model(X,theta) :**

-Elle prend comme paramètre la matrice X et un vecteur thêta

-Cette fonction retourne une matrice colonne qui contient les prédictions

2/Créer une fonction **grad(X,y,theta):**

Cette fonction retourne la matrice colonne des dérivées partielles

3/Créer une fonction **descente_grad(X,y,theta,pas,iterations):**

-Cette fonction implémente l'algorithme de descente de gradient

-elle retourne le vecteur theta final après l'actualisation des paramètres : b1 et b0

4/ la visualisation du résultat : nuage des points + droite de régression

5/ Donner le coefficient de détermination du model

6/ Vérification en utilisant la bibliothèque sklearn

```
from sklearn.linear_model import LinearRegression

Regression_model=LinearRegression()

Regression_model.fit(x,y)

Regression_model.score(x,y)
```

En principe le score retourné = $[\text{Corrcoef}(x,y)]^2$

Partie 02 : Regression multiple : $f(x) = b_0 + \sum_{j=1}^{p} b_j x_j$ ***avec ,*** $\vec{B} \in R^{p+1}$ **.**

Pour le jeu de données D={($\vec{x}$i,yi)},on peut utiliser le module suivant pour la génération du D

from sklearn.datasets import make_regression

x,y=make_regression(n_samples=100,n_features=5,noise=3)

Les variables :

La variable y : la matrice colonne des étiquettes

La variable X : une matrice des observations (avec une colonne des 1 à la fin)

Solution 01:

$$\vec{\beta}^* = (X^T X)^{-1} X^T \vec{y}$$

Solution 02 : adapter la solution précédente (simple) pour résoudre le problème de la régression multiple en utilisant la méthode de descente de gradient

Passage à la réalité

En pièce jointe un dataset qui concerne le marché des automobiles chinois

Le dataset contient essentiellement deux informations:

-des information sur l'automobile (id, marque, puissante, ,,,,,) : les variables explicatives

-le prix de vente : variable à expliquer (à prédire)

Travail demandé :

Construire un modèle de Regression (linéaire) qui permet de prédire le prix de vente d'une nouvelle voiture

Démarche :

1- Nettoyez les données : assurer qu'elles sont consistantes, sans valeurs aberrantes ni manquantes (

2- Explorez les données : identifier les variables explicatives pertinentes, qui ont vraiment un impact sur le prix

3- Préparer les données pour la phase de modélisation : penser à transformer les données (catégories) en données numériques

4-Modelisation : modèle de régression linéaire :

-diviser les données en deux partie (train_set,test_set)

-entrainer le modèle sur train_set ensuite tester le sur test_set

-L'objectif est d'avoir un meilleur score (tout dépend de choix des variables, préprocessing ….)