Ecole Centrale Casablanca

Apprentissage Statistique et Réseaux de Neurones

TP 1 : Régression linéaire

Auteur :
Mohamed EL MAIMOUNI

17 octobre 2020

Nomenclature

σ_β	Vecteur de $\mathbb{R}^p$
T	Vecteur de $\mathbb{R}^p$
X	Matrice de dimension n x p
β	Vecteur de $\mathbb{R}^p$
ϵ	Bruit blanc gaussien de variance 2

Introduction

Ce rapport synthétise et analyse les résultats obtenus à travers le premier TP du module "**Apprentissage statistique et réseaux de neurones**".

La première partie de ce rapport consiste en une étude théorique de la régression linéaire évoquant les différents facteurs induits par une régression. Dans la deuxième partie, nous allons prendre un jeu de données réelles, et y effectuer une régression linéaire.

Les valeurs simulées seront données à 5 chiffres significatifs

1 Étude théorique

Dans cette partie, nous allons construire un jeu de données puis appliquer une régression linéaire et étudier les différentes valeurs données par cette régression.
Soit alors X notre jeu de données d'entrée : en vrai vie, les différents éléments desquels dépend la grandeur qu'on veut modéliser, sera notée Y dans la suite.

Génération du modèle

Afin de générer une matrice (n,p) de plein rang, on propose d'utiliser la fonction $matrix()$ de **R** avec :

— $data = randomNumbers(n = n, min = -n/2, max = n/2+1, check = TRUE)$, (pour ne générer que des positifs on peut fixer $(min, max) = (1, n + 1)$)
— $nrow = n$
— $ncol = p$
— $byrow = TRUE$, pour remplir ligne par ligne

Pour s'assurer que la matrice est de plein rang, on va utiliser la logique suivante : *r <- 0*
while(r != p) {
X = matrix(...)
r = qr(X)$rank }

On peut s'assure que le vecteur colonne (1,1,..,1) n'est pas dans l'image de X en ajoutant ce vecteur à X et s'assurer que le rang de X est p+1

On génère un bruit blanc gaussien à travers une distibution normale de paramètres (0,σ^2), où $\sigma^2 = 2$. Dans le language **R**, on peut utiliser la fonction $rnorm()$ avec :

— $n = n$: nombre d'éléments de ce vecteur
— $mean = 0$
— $sd = sqrt(2)$

À ce stade, nous avons générer une matrice $X \in \mathrm{M}_{n,p}(\mathbb{R})$, un bruit gaussien $\epsilon \in \mathbb{R}^p$. On fixe $\beta = (-2, 7), n = 100$ *et* $p = 2$. Compte tenu de ces données et paramètres on génère un vecteur $Y \in \mathbb{R}^n$

$$Y = X\beta + \epsilon$$

Les figures suivantes sont des plot de Y en fonction des colonnes de X selon deux scénarios : X est remplie de coefficients positifs ou de coefficients positifs et négatifs

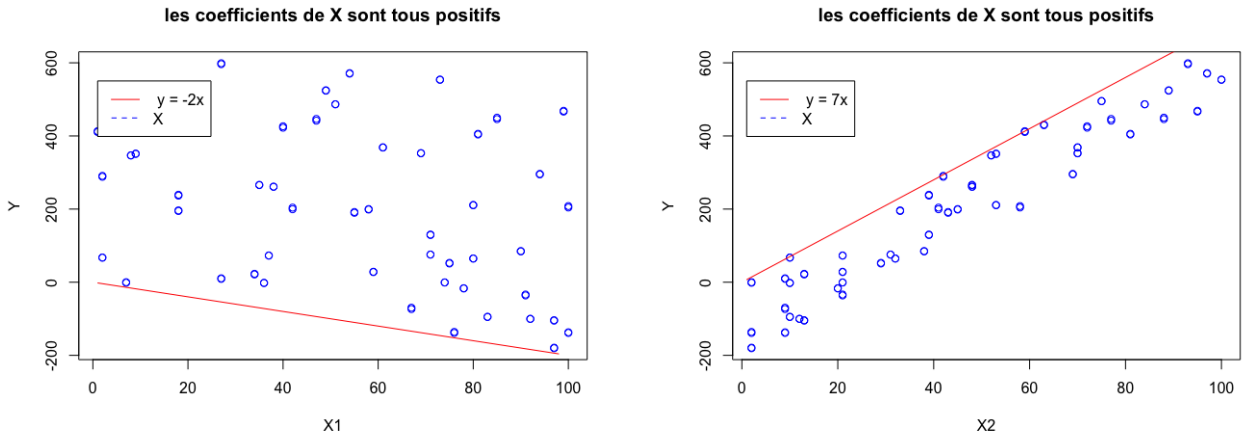

FIGURE 1 – Plot de Y par rapport à X_1 *et* X_2(les éléments de X sont tous positifs)

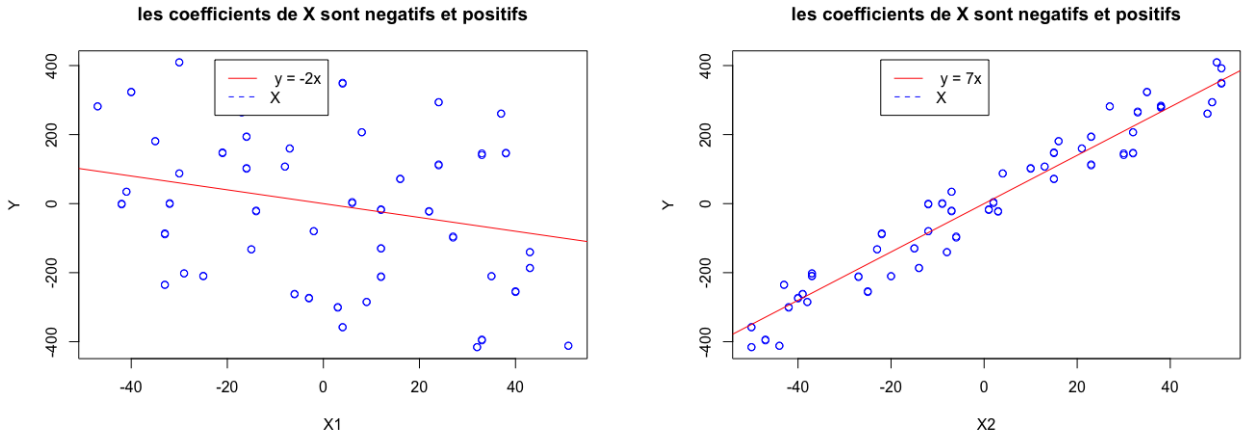

FIGURE 2 – Plot de Y par rapport à X_1 *et* X_2(X est remplie d'éléments positifs et négatifs)

Analyse et discussion des plots

Dans les différentes figures on remarque la présence d'une droite rouge. L'équation de cette droite est :

$$y = \alpha x$$

Où

$$x = \frac{max(X_i) - min(X_i)}{n + max(X_i)} * C$$

avec $i \in \{1, 2\}$, $\alpha \in \{-2, 7\}$ et $C \in \mathbb{M}_{1,p}([min, max])$, *min et max de randomNumbers (cf. section 1)*

La relation ci-dessus permet d'avoir des éléments dans l'axe abscisses de même ordre de grandeur que ceux des différentes colonnes de X, d'où la possibilité de d'avoir le tracé de la droite $y = \alpha x$ et les plots (X_i, Y) dans une même figure

De plus, le coefficient directeur de cette droite est α, et nous avons fixé $\beta_i = \alpha$, on peut dire que alors que la droite rouge est une régression linéaire des couples de données (Xi, Y). Mais dans le cas où les coefficients de X sont tous positifs, on remarque que la droite semble être translatée vers le bas ! ! Peut-être il manque une ordonée à l'origine de la droite ?

Dans la suite, nous allons travailler avec une matrice d'autant de coefficients positifs que négatifs.

Calcul de $\hat{\beta}$

Dans cette partie, on va calculer $\hat{\beta}$, l'estimateur de β, à partir des observations X et Y simulées. D'après le cours de la régression linéaire $\hat{\beta}$ est peut-être calculée à partir de la relation suivante :

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

En implémentant cette relation dans R studio, on trouve $\hat{\beta} = (-2.00518, 7.00114)$. On remarque que l'estimateur de β est différent de β de quelques centièmes : c'est l'effet des approximations numériques que fait **R** lors des calcul des produits matriciels et des inverses des matrices.

La fonction lm

En utilisant la fonction *lm* de **R**, on peut calculer $\hat{\beta}$. En implementant cette fonction dans R studio avec nos observations simulées (X, Y), on trouve que $\hat{\beta} = (-0.40032, -2.00525, 7.00101)$.

Dans la suite, $X \in \mathbb{R}^{nx(p+1)}$, on va ajouter X un vecteur colonne $(1, 1, .., 1) \in \mathbb{R}^n$. On recalcule $\hat{\beta}$ comme dans la section n° 2, on trouve $\hat{\beta} = (-0.40032, -2.00525, 7.00101)$ L'ajout de la colonne des 1 à la matrice X nous a permis d'avoir l'ordonnée à l'origine (aka intercept β_0), on remarque que notre $\hat{\beta}$ simulée est égale à celle calculée par la fonction *lm*.

Calcul des autres grandeurs...

Ècart-type du bruit gaussien

$\hat{\sigma}$ est l'écart-type du bruit gaussien(Residual standard error), donné par :

$$\hat{\sigma^2} = \frac{||Y - \hat{Y}||^2}{n - p}$$

Où $\hat{Y}$ est le vecteur des valeurs prévues
Après le calcul, nous avons $\hat{\sigma^2} = 1.71415$ donc $\hat{\sigma} = 1.30925$. En outre, la fonction *summary()* nous a retourné un *Residual standard error* : 1.316 on 97 degrees of freedom.

Coefficient de corrélation empirique

R^2 est le coefficient de corrélation empirique entre les valeurs prévues $\hat{Y}$ et les valeurs observées Y, donné par :

$$R^2 = \frac{||\hat{Y} - \bar{Y}||^2}{||Y - \bar{Y}||^2}$$

Où $\bar{Y} = \frac{1}{n}\Sigma_{i=1}^{n} Y_i$, est la moyenne empirique de Y

Le calcul à travers l'implémentation dans **R**, on trouve $R^2 = 0.99996$. La fonction summary() a retourné un *Multiple R-squared* égale à 1

Estimateur de l écart-type de l'estimateur $\hat{\beta}$

$\hat{\sigma_\beta}$ est un estimateur de l'écart-type de $\hat{\beta}$, c'est un vecteur de $\mathbb{R}^p$, donné par :

$$\hat{\sigma_{\beta_j}} = \hat{\sigma^2}[(X^t X)^{-1}]_{jj}$$

Notre grandeur simulée est $\hat{\sigma_\beta} = (0.131606, 0.0049666, 0.0042935)$, celle donnée la fonction *summary()* est : $(0.13160, 0.004967, 0.004294)$

Statistique de test

On appelle statistique de test, la variable aléatoire T_i qui nous permet de rejeter l'hypothèse null $H_0 : \beta_i = 0$ *si* $|T_j| > t$, où t est donnée par $P(|T_j| > t) = \alpha$, avec α le niveau de test.
T_j est donnée par :

$$T_j = \frac{\hat{\beta}_j}{\hat{\sigma_{\beta_j}}}$$

La fonction summary() nous a retourné $(-3.042, -403.743, 1630.574)$, et le vecteur que nous avons calculé est : $(-3.0418, -403.7434, 1630.574)$.
Or les données affichées par la fonction *summary()* indiquent que pour $j = 0$: $\mathbf{Pr(>|t|)} = 0.00302 < 0.05$, on peut alors rejeter l'hypothèse H_0, Autrement dit, $\beta_0 \neq 0$, ce qui est en parfaite adéquation avec notre $\hat{\beta}_0$

Statistique du test de Fisher

La statistique du test de Fisher est donnée par :

$$F = \frac{R^2}{1 - R^2} \frac{n - p}{p - 1}$$

Elle suit la loi de Fisher de paramètre $(p - 1, n - p)$ sous $H_0 : \beta_j = 0, \forall j \neq 1$
La fonction *summary()* a retourné une valeur de $F = 1.507\text{x}10^6$, avec un **p-value** $< 2.2e - 16$ et nous avons calculé une valeur de $F = 1.50662\text{x}10^6$
La valeur de **p-value** étant inférieur à 0.05, on peut alors rejeter l'hypothèse H_0

2 Données réelles

2.1 Boston Housing Data

Dans cette partie, nous allons travailler sur la base données : **Boston Housing Data**.
Cette base de données modélise 14 différents aspects des banlieues de Boston. Avec 506 observations.
Nous allons modéliser le nombre de crime par habitants en fonction des différentes informations de la base de données.

Le tableau suivant résume les différentes informations contenues dans notre jeu de données :

Critère	Description
CRIM	Taux de criminalité par habitant par ville
ZN	Proportion des terrains résidentiels zonés pour les lots de plus de 25 000 pieds carré
INDUS	Proportion des surfaces commerciales non commerciales par ville
CHAS	Variable fictive de la rivière Charles (= 1 si le tronçon délimite la rivière ; 0 sinon)
NOX	Nitric oxides concentration (parts per 10 million).
RM	Nombre moyen de pièces par logement
AGE	proportion des logements occupés par leurs propriétaires construits avant 1940
DIS	Les distances pondérées par rapport aux cinq centres d'emploi de Boston.
RAD	Indice d'accessibilité aux autoroutes radiales
TAX	Taux d'impôt foncier sur la valeur totale par 10 000 $
PTRATIO	Ratio élèves-enseignants par ville
B	$1000(B_k - 0,63)^2$ où B_k est la proportion de noirs par ville
LSTAT	% de statut inférieur de la population
MEDV	Valeur médiane des logements occupés par leurs propriétaires en milliers de dollars

Régression linéaire

À travers la fonction *lm* de R, nous allons effectuer une régression linéaire du taux de criminalité par habitant par ville en fonction des autres informations de la base de données.
Le tableau suivant récapitule le résultat de cette régression :

Critère	Description
R^2	0.454
Intercept	17.03322
$Pr(>\|t\|)$ de *Intercept*	0.01894
PTRATIO	-0.271081
$Pr(>\|t\|)$ de *PTRATIO*	0.146611
p-value du test de Fisher	$<$ 2.2e-16

On remarque d'abord que notre R^2 est inférieur à 0.5, on peut alors dire que notre modèle est loin d'être bon pour bien modéliser le taux de criminalité, de plus p-value < 0.5, on peut dire que notre modèle est mieux que

de ne considérer que l'ordonnée à l'origine (modèle constant). Toutefois, la valeur obtenue de l'ordonnée à l'origine est statistiquement significative (car $Pr(>|t|) < 0.05$).
Enfin, le le facteur β_j associé au ratio élèves-enseignement n'est pas significativement non nul, car $Pr(>|t|) > 0.05$, on ne peut pas alors rejeter l'hypothèse nulle

Matrice de corrélation

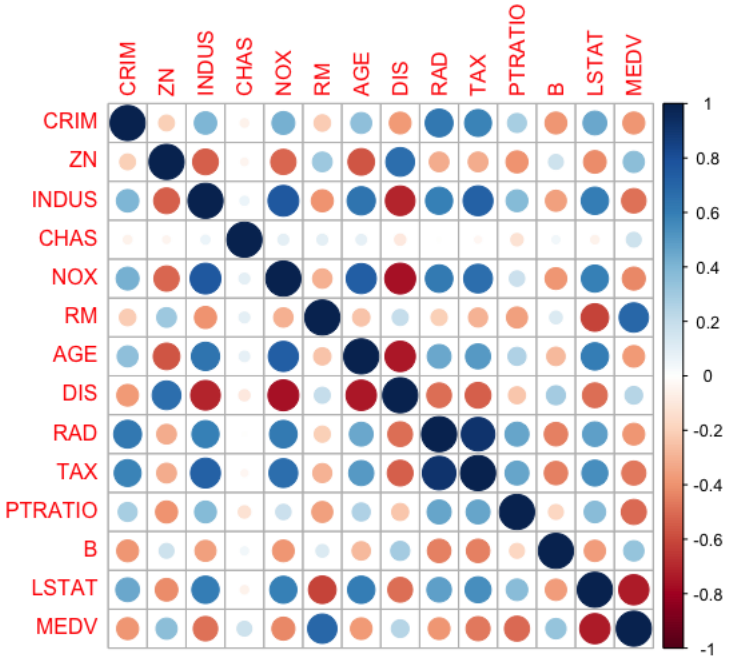

FIGURE 3 – Matrice de corrélation

On remarque que la variable qu'on veut modéliser est corrélée avec quelques critères et qu'elle ne l'ai pas du tout avec CHAS, on peut alors se permettre de la supprimer pour affiner notre modèle (Réduction de la complexité du modèle)

2.2 Forest Fires

Nous allons étudier la base de donnée **Forest Fires**, pour plus de détails sur ce jeu de données : Lisez le fichier forestfires.names

Nous allons alors essayer de modéliser la variable **area** en fonction des autres via une régression linéaire par la fonction `lm`

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -15.16402   76.56086  -0.198   0.8431
X             2.25583    1.49786   1.506   0.1327
Y            -0.14765    2.81881  -0.052   0.9582
monthaug     46.88205   38.08792   1.231   0.2190
monthdec     47.37821   36.94830   1.282   0.2004
monthfeb      5.58985   25.94816   0.215   0.8295
monthjan     14.76909   56.40617   0.262   0.7936
monthjul     28.87889   33.05232   0.874   0.3827
monthjun      6.71548   30.33765   0.221   0.8249
monthmar     -4.22256   23.41447  -0.180   0.8570
monthmay     12.79646   50.91572   0.251   0.8017
monthnov     -4.41010   68.37767  -0.064   0.9486
monthoct     68.97536   45.42009   1.519   0.1295
monthsep     73.73192   42.67672   1.728   0.0847 .
daymon        5.96928   10.48154   0.570   0.5693
daysat       19.40993   10.06218   1.929   0.0543 .
daysun        5.14460    9.78870   0.526   0.5994
daythu        9.67192   11.10696   0.871   0.3843
daytue        7.79282   10.88291   0.716   0.4743
daywed        5.47914   11.40526   0.480   0.6312
FFMC         -0.09527    0.76985  -0.124   0.9016
DMC           0.20106    0.08681   2.316   0.0210 *
DC           -0.12880    0.05872  -2.194   0.0287 *
ISI          -0.54416    0.83105  -0.655   0.5129
temp          1.29620    1.03082   1.257   0.2092
RH           -0.13476    0.28845  -0.467   0.6406
wind          1.97427    1.77824   1.110   0.2674
rain         -2.81545    9.92647  -0.284   0.7768
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 63.88 on 489 degrees of freedom
Multiple R-squared:  0.04578,   Adjusted R-squared:  -0.006905
F-statistic: 0.8689 on 27 and 489 DF,  p-value: 0.6581

>
```

FIGURE 4 – Résumé de la fonction lm

La première remarque qu'on doit faire est que notre modèle est **médiocre**, en effet, avec un $R^2 = 0.04578$ et la plupart des coefficients ont $Pr(> |t|) > 0.05$ ce qui rend ces derniers sans signification statistique

De plus, **p-value** $= 0.6581$ du test de Fisher > 0.05, alors le modèle constant (ne prendre en considération que l'ordonnée à l'origine, pas celui affiché dans le résumé car son $Pr(> |t|) > 0.05$) est mieux que le notre

Un dernier constat est l'apparition de plusieurs coefficients qui n'existent pas dans la liste de nos critères (monthaug, monthfeb, daymon, daytue). En effet, les colonnes day et month sont catégorielles, et la régression linéaire ne traite

que les données numériques , on dit alors que la fonction `lm` a encodé ces deux critère.

Régression de log(1+area)

Pour affiner notre modèle, on peut appliquer des transformations aux variables $(X^2, log(1+X), ..etc)$
Nous allons donc appliquer la transformation $X \rightarrow log(1+X)$ à notre variable de sortie (area)

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.5705460  1.6566550  -0.344  0.73070
X            0.0524204  0.0324114   1.617  0.10645
Y           -0.0184700  0.0609946  -0.303  0.76216
monthaug     0.3274391  0.8241619   0.397  0.69132
monthdec     2.2050797  0.7995023   2.758  0.00603 **
monthfeb     0.1886078  0.5614767   0.336  0.73708
monthjan    -0.3163816  1.2205397  -0.259  0.79558
monthjul     0.0991694  0.7151995   0.139  0.88978
monthjun    -0.2862231  0.6564584  -0.436  0.66302
monthmar    -0.3416243  0.5066518  -0.674  0.50045
monthmay     0.7175267  1.1017352   0.651  0.51518
monthnov    -1.1031443  1.4795838  -0.746  0.45628
monthoct     0.8232625  0.9828184   0.838  0.40263
monthsep     0.9934196  0.9234562   1.076  0.28256
daymon       0.1457734  0.2268038   0.643  0.52070
daysat       0.3099153  0.2177296   1.423  0.15526
daysun       0.2109897  0.2118118   0.996  0.31969
daythu       0.0722394  0.2403369   0.301  0.76387
daytue       0.3222933  0.2354888   1.369  0.17175
daywed       0.1978808  0.2467916   0.802  0.42305
FFMC         0.0074547  0.0166582   0.448  0.65471
DMC          0.0041790  0.0018785   2.225  0.02656 *
DC          -0.0020052  0.0012706  -1.578  0.11516
ISI         -0.0147970  0.0179825  -0.823  0.41099
temp         0.0360374  0.0223054   1.616  0.10682
RH           0.0006673  0.0062416   0.107  0.91490
wind         0.0603127  0.0384782   1.567  0.11766
rain         0.0309440  0.2147931   0.144  0.88551
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.382 on 489 degrees of freedom
Multiple R-squared:  0.07426,   Adjusted R-squared:  0.02315
F-statistic: 1.453 on 27 and 489 DF,  p-value: 0.06765

>
```

FIGURE 5 – Résumé de la fonction lm

Le premier constat qu'on peut faire est l'amélioration du R^2 (***Adjusted R-squared*** aussi), on dit alors que notre modèle est amélioré. Cependant, le **p-value** du test de Fisher reste supérieur à 0.05.

De plus, la plupart des coefficients restent sans aucune signification statistique (leur $Pr(>|t|) > 0.05$), seul **DMC** qui a eu une p-value du T-test $<$ 0.05 dans les deux modèle. On dit alors que *The Duff Moisture Code* est le facteur le plus important

Conclusion

Dans ce TP, nous avons énuméré les différents facteurs qui définissent la performance d'une régression linéaire, à savoir R^2, **p-value**, $Pr(>|t|)$, $..etc$. Puis nous avons appliqué cette régression à des jeux de données réelles et nous avons interprété les résultats à l'aide des facteurs décrit dans la partie théorique. Enfin, nous avons mentionné l'importance de faire une transformations pour des variable en guise d'améliorer le modèle.

Un script en **R** permettant de tester et d'appliquer ce qui a été fait dans ce rapport sera joint à ce document.