



**RAPPORT FINAL FOUILLE  
DE DONNEES SUR LA  
PRATIQUE D'UNE METHODE  
D'APPRENTISSAGE  
SUPERVISEE AU CHOIX**

**Groupe 6 :**

- **MEDOU DANIEL MAGLOIRE, P21**
- **NGASSA TCHOUDJEUH TATIANA, P20**

**Etudiant en Master 1, option SIM**

**Sous la supervision de :**

**Dr. NGUYỄN Thị Minh Huyền**

**Année académique  
2016-2017**

# TABLE DES MATIERES

TABLE DE FIGURES .....	3
INTRODUCTION .....	4
I. CHOIX D'UNE METHODE D'APPRENTISSAGE SUPERVISE .....	5
1. Choix de la méthode .....	5
2. Présentation de la Régression Logistique Binaire .....	5
II. PREPARATION DES DONNEES.....	7
1. Données.....	7
2. Prétraitement des données.....	8
3. Traitement des valeurs manquantes .....	8
4. Choix d'un ensemble d'entraînement et de validation.....	9
III. CHOIX DES PARAMETRES ET APPLICATION DE LA METHODE AU JEU DE DONNEES.....	9
1. Choix des paramètres.....	9
2. Application des paramètres à la méthode d'apprentissage supervisé .....	11
a. Modèle du « Test1 » avec les attributs numériques.....	11
b. Modèle du « Test2 » avec les attributs numériques.....	13
c. Modèle du « Test3 » avec les attributs numériques.....	15
d. Modèle du « Test4 » avec les attributs numériques.....	16
3. Prédiction : Evaluation de la capacité d'un modèle .....	19
a. Prédiction du modèle « ModeleNum » des attributs numériques .....	19
b. Prédiction du modèle « ModeleTest2 » des attributs numériques .....	21
c. Prédiction du modèle « ModeleTest3 » des attributs numériques .....	22
d. Prédiction du modèle « ModeleTest3 » des attributs numériques .....	23
IV. CLASSIFICATION PAR ARBRE DE DECISION .....	23
1. Construction d'un arbre de décision des attributs numériques.....	24
a. Test1 : Construction du modèle « Modele_Arbre » .....	24
b. Prédiction et interprétation des résultats .....	24
c. Matrice de confusion du test1.....	25
d. Matrice de confusion du test2.....	25
e. Matrice de confusion du test3.....	25
V. COMPARAISON DES RESULTATS DES DEUX METHODES D'APPRENTISSAGE SUPERVISEES.....	26
CONCLUSION.....	27
Références .....	28

## TABLE DE FIGURES

Figure 1: Corrélation entre différentes paires de variables .....	10
Figure 2: Modèle des cinq attributs numériques.....	11
Figure 3: Modèle des attributs numériques avec suppression d'un.....	12
Figure 4: Choix du meilleur modèle des attributs numériques.....	12
Figure 5: Odd-ratio du modèle "ModeleNum" .....	13
Figure 6: ModeleTest2 .....	13
Figure 7: ModeleTest2bis .....	14
Figure 8: Valeurs AIC des attributs du Modèle Test2 .....	14
Figure 9: ModeleTest3 .....	15
Figure 10: ModeleTest3bis .....	15
Figure 11: Valeurs AIC des attributs du Modèle Test3 .....	16
Figure 12: Modele "Test4" .....	17
Figure 13: Modèle des attributs catégoriels .....	18
Figure 14: Choix du meilleur modèle des attributs discrets .....	19
Figure 15: Probabilités des attributs numériques du modèle "ModeleNum" .....	20
Figure 16: Probabilité du modèle "ModeleNum" selon le seuil défini à 0.5 .....	20
Figure 17: Matrice de confusion du modèle "ModeleNum" .....	21
Figure 18: Estimation des taux de bonne classification et de mauvaise classification du « ModeleNum ».....	21
Figure 19: Matrice de confusion du modèle "ModeleTest2" .....	21
Figure 20: Estimation de taux de bonne classification et de mauvaise pour le "ModeleTest2" ..	22
Figure 21: Matrice de confusion du modèle "ModeleTest3" .....	22
Figure 22: Estimation du taux de bonne classification et de mauvaise du modèle "ModeleTest3" .....	22
Figure 23: Matrice de confusion "Modele Test4" .....	23
Figure 24: Estimation du taux de bonne classification et de mauvaise du modèle « ModeleTest4 » .....	23
Figure 25: Modèle pour la construction de l'arbre .....	24
Figure 26: Prédiction du modèle "Modele_Arbre".....	24
Figure 27: Matrice de confusion du modèle "Modele_Arbre" .....	25
Figure 28: Matrice de confusion du modèle "Modele_Arbre2" .....	25
Figure 29: Matrice de confusion du modèle "Modele_Arbre3" .....	25
Figure 30: Matrice de confusion du modèle "Modele_Arbre4" .....	26

## INTRODUCTION

L'apprentissage statistique ou apprentissage automatique (Machine Learning en anglais) ou encore Fouille de Données (Data Mining) comprend différentes méthodes de recherche d'informations dans un jeu de données à des fins prévisionnelles et/ou et décisionnelles. L'apprentissage supervisé étant une technique qui nous permettra à produire automatiquement des règles à partir d'une base de données d'apprentissage. Pour ce faire, nous avons plusieurs méthodes d'apprentissage supervisé mise à notre disposition à savoir **Boosting, Machine à Secteurs de Support, Réseau de neurones, Méthode de K plus proches voisins, Arbre de décision, Classification naïve bayésienne** et bien d'autres existant. Toutes ces méthodes ne sont pas appliquées à un jeu de données d'apprentissage comme bon nous semble car plusieurs critères entrent en jeu pour appliquer une quelconque à un jeu de données et selon l'objectif ou le but de ce dernier.

# I. CHOIX D'UNE METHODE D'APPRENTISSAGE SUPERVISE

## 1. Choix de la méthode

Le choix d'une méthode d'apprentissage supervisé à appliquer sur un jeu de données, étant fonction de certains critères selon lesquels, l'objectif du jeu de données et le type de variables que nous disposons dans ce dernier nous amène à opérer un choix sur une méthode ou une classification d'apprentissage automatique.

Dans notre cas d'espèce, nous avons jeté notre dévolu sur la **REGRESSION LOGISTIQUE BINAIRE**. Pourquoi ce choix ?

- La Régression Logistique, est tout d'abord une méthode qui fait appel aux attributs continus et/ou discrets ;
- L'objectif de notre jeu de données étant **de prédire ou de déterminer si une personne fait plus de 50k par année** et cette variable de prédiction est divisée en deux classes  $>50K$  et  $\leq 50K$ .
- L'objectif est de prédire les valeurs prise par la variable aléatoire  $Y$  définie dans  $\{y_1, y_2, \dots, y_n\}$ . Pour la Régression Logistique Binaire,  $Y$  prend uniquement deux modalités  $\{+, -\}$  ou  $\{0, 1\}$

A partir de ces trois critères voir les deux premiers, nous avons choisi d'appliquer la **REGRESSION LOGISTIQUE BINAIRE** à notre jeu de données car, cette méthode vise à expliquer une variable dite binaire.

## 2. Présentation de la Régression Logistique Binaire

La régression logistique a pour but, l'explication de la variable qualitative  $Y$  en fonction de variables quantitatives. L'idée est d'utiliser un modèle linéaire généralisé pour modéliser la probabilité d'appartenance de  $Y$  à une modalité en fonction des variables explicatives.

Dans le cas du **modèle binomial** :

Soit  $Y = (Y_1, \dots, Y_n)^T$  un vecteur aléatoire à valeurs dans  $\{0; 1\}^n$ . On observe une réalisation  $Y = (Y_1, \dots, Y_n)^T$  appartenant  $\{0; 1\}^n$  de  $Y$  et pour chaque  $Y_i$ , un ensemble de variables  $X_i = (x_{1i}, \dots, x_{qi})$ . On suppose pour simplifier que la constante est une variable du modèle en posant  $x_{1i} = 1$ . Les variables  $X_{ji}$  peuvent être quantitatives ou qualitatives, auquel cas elles sont exprimées sous la forme d'indicatrice. Les données consistent donc en l'observation de  $Y$  et de la matrice

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} x_{11} & x_{21} & \dots & x_{k1} \\ x_{12} & x_{22} & \dots & x_{k2} \\ \vdots & \vdots & \vdots & \vdots \\ x_{1n} & x_{2n} & \dots & x_{qn} \end{bmatrix} \in \mathbb{R}^{n \times q}.$$

Pour  $i = 1, \dots, n$  on note  $\pi_i = P(Y_i = 1)$  et  $\pi = (\pi_1, \dots, \pi_n)^T$  appartenant à  $(0; 1)^n$ . La régression logistique est basée sur l'hypothèse que les probabilités  $\pi_i$  vérifient

$$\pi_i = \frac{e^{\mathbf{x}_i \beta}}{1 + e^{\mathbf{x}_i \beta}}, \quad i = 1, \dots, n$$

où  $\beta = (\beta_1, \dots, \beta_q)^T$  appartenant à  $\mathbb{R}^q$  est un vecteur inconnu qui ne dépend pas de l'indice  $i$ . Cette hypothèse est une façon de modéliser la dépendance entre la variable binaire  $Y_i$  et le vecteur de variables explicatives  $\mathbf{X}_i$ . Le choix de la transformation  $x \mapsto e^x / (1 + e^x)$  est un moyen simple de se restreindre à des valeurs dans  $(0; 1)$  pour évaluer la probabilité  $\pi_i$ . Même si une relation exactement de cette forme est peu probable en réalité, le modèle est suffisamment flexible pour espérer s'en approcher (cette observation est aussi valable pour le modèle linéaire général). La fonction réciproque est la fonction **logit**:

$$\text{logit}(p) = \ln \left( \frac{p}{1 - p} \right), \quad p \in (0, 1).$$

Le modèle de la régression logistique peut donc s'écrire

$$\text{logit}(\pi_i) = \mathbf{x}_i \beta = \sum_{j=1}^q x_{ji} \beta_j, \quad i = 1, \dots, n.$$

Les probabilités  $\pi_i$  ne sont bien sûr pas connues, mais on observe les valeurs « bruitées »  $Y_i = \pi_i + \epsilon_i$  où les bruits  $\epsilon_i$  sont centrés, de loi

$$\mathbb{P}(\epsilon_i = 1 - \pi_i) = \pi_i \quad \text{et} \quad \mathbb{P}(\epsilon_i = -\pi_i) = 1 - \pi_i.$$

On a donc

$$y_i = \frac{e^{\mathbf{x}_i \beta}}{1 + e^{\mathbf{x}_i \beta}} + \epsilon_i,$$

ce qui fait de la régression logistique un cas particulier du modèle linéaire généralisé. En supposant l'indépendance des  $Y_i$ , la vraisemblance du modèle est donnée par

$$\mathcal{L}(y, b) = \mathbb{P}(Y = y | \beta = b) = \prod_{i=1}^n \left( \frac{e^{\mathbf{x}_i b}}{1 + e^{\mathbf{x}_i b}} \right)^{y_i} \left( 1 - \frac{e^{\mathbf{x}_i b}}{1 + e^{\mathbf{x}_i b}} \right)^{1 - y_i} = \prod_{i=1}^n \frac{e^{\mathbf{x}_i b y_i}}{1 + e^{\mathbf{x}_i b}}, \quad b \in \mathbb{R}^q.$$

Une forme analytique de l'estimateur du maximum de vraisemblance  $\hat{\beta}$  n'étant en général pas disponible, on peut l'approcher par des méthodes numériques. On obtient alors une estimation des probabilités  $\pi_i$  par

$$\hat{\pi}_i = \frac{e^{\mathbf{x}_i \hat{\beta}}}{1 + e^{\mathbf{x}_i \hat{\beta}}}, \quad i = 1, \dots, n.$$

Si  $\hat{\beta}$  converge en probabilité vers  $\beta$  (ce qui peut se montrer sous des hypothèses raisonnables sur les  $\mathbf{X}_i$ ), alors on peut déduire le comportement asymptotique de  $\hat{\beta}$ . On vérifie en effet facilement que les conditions du premier ordre donnent

$$0 = \frac{d}{db} \ln \mathcal{L}(y, \hat{\beta}) = \mathbf{X}^\top (y - \hat{\pi})$$

Avec  $\hat{\pi} = (\hat{\pi}_1, \dots, \hat{\pi}_n)^\top$ . Par un développement limité d'ordre 1, on obtient

$$\hat{\pi}_i = \frac{e^{\mathbf{x}_i \hat{\beta}}}{1 + e^{\mathbf{x}_i \hat{\beta}}} = \pi_i + \frac{\pi_i}{1 + e^{\mathbf{x}_i \beta}} \mathbf{x}_i (\hat{\beta} - \beta) + o\|\hat{\beta} - \beta\|.$$

Soit  $\Sigma = \text{var}(y)$ , c'est-à-dire  $\Sigma_{ii} = \pi_i/(1 + e^{\mathbf{x}_i \beta})$  et  $\Sigma_{ij} = 0$  pour  $i \neq j$ . On déduit

$$\hat{\beta} - \beta \approx (\mathbf{X}^\top \hat{\Sigma} \mathbf{X})^{-1} \mathbf{X}^\top (y - \pi). \quad (1)$$

Si  $n \mathbf{X}^\top \hat{\Sigma} \mathbf{X}$  converge vers une matrice  $B$  quand  $n \rightarrow \infty$  et  $\sqrt{n} \mathbf{X}^\top (y - \pi)$  est asymptotiquement Gaussien (comme c'est le cas par exemple si les  $\mathbf{x}_i$  sont iid et de carré intégrable), alors

$$\sqrt{n}(\hat{\beta} - \beta) \xrightarrow[n \rightarrow \infty]{} \mathcal{N}(0, B^{-1}).$$

## II. PREPARATION DES DONNEES

### 1. Données

Nous utiliserons le même jeu de données « **Adult** » comportant **32 561** observations et **14** variables prédictives pour illustrer la Régression Logistique Binaire. L'objectif étant de prédire si une personne fait plus de 50K par année (PREDICTION – Y ; avec “<=50” = “non” et “>50” = “oui”) à partir de son AGE (quantitative –  $X_1$ ), du WORKCLASS (qualitatif –  $X_2$ ), de son EDUCATION (qualitatif –  $X_3$ ), de son OCCUPATION (qualitatif –  $X_4$ ), de son MARITAL-STATUT (qualitatif –  $X_5$ ), de son RELATIONSHIP (qualitatif –  $X_6$ ) et de son HOURS-PER-WEEK (quantitatif –  $X_7$ ).

## 2. Prétraitement des données

Dans le monde réel, les données proviennent de plusieurs sources et de différents processus. Par conséquent, elles peuvent contenir des anomalies ou bien des valeurs incorrectes qui compromettent la qualité du jeu de données. Les problèmes les plus fréquents liés à cette qualité sont :

**Caractère incomplet** : les valeurs ou les attributs sont manquants ;

**Bruit** : les données contiennent des valeurs erronées ou des aberrations ;

**Incohérence** : les données contiennent des enregistrements en conflit.

La qualité de données est essentielle pour obtenir des modèles prédictifs importants. Notre jeu de données a été analysé et comme résultat de notre analyse, nous avons **32 561 nombres d'enregistrements, 15 attributs parmi lesquels 7 sont continus et 8 attributs nominaux. Enfin plusieurs valeurs manquantes.**

**NB :** Nous allons, dans notre jeu de données créer une nouvelle colonne « salaire » qui sera binaire avec ' $\leq 50$ ' = '0' et ' $> 50$ ' = '1'.

## 3. Traitement des valeurs manquantes

Elles sont des valeurs d'une ou plusieurs variables non observés chez les individus. Elles peuvent provenir des données jamais enregistrées, perdues, effacées ou de fausses données. Si elles sont en faibles nombres, elles peuvent être simplement ignorées et si elles sont en plus grand nombre et que les ignorer peut entraîner une forte perte d'informations, il existe d'autres moyens pour leur attribuer une autre valeur (imputation). **L'imputation** est la manière utilisée dans notre jeu de données de **32 561 observations** pour pallier aux différentes valeurs manquante. Traitée sous l'outil R qui nous donne la possibilité par le biais d'un fichier script R contenant les commandes suivantes ceci pour le traitement des valeurs manquantes :

```
> longueur=length(Adult_Data$Nom_Attribut)
> for(i in 1: longueur)
+ {}
> longueur=length(Adult_Data$ Nom_Attribut)
> for(i in 1: longueur)
+ {
+ Adult_Data$ Nom_Attribut [is.na(Adult_Data$ Nom_Attribut)]<-sample(Adult_
Data$ Nom_Attribut, sum(is.na(Adult_Data$ Nom_Attribut)))}
```

**NB :** Nous avons essayé l'imputation par régression. Celle-ci prenait beaucoup de temps pour s'exécuter sur nos machines vue le nombre important des données; faute de temps nous avons opté pour la méthode aléatoire.



## 4. Choix d'un ensemble d'entraînement et de validation

Notre jeu de données a été divisé en deux parties, constituant un ensemble de données d'entraînement et un autre pour les données de validation

**Test1** : Cas du sur apprentissage ; données d'entraînement (99%) supérieur aux données de validation (1%).

- Les données d'entraînement sont comprises entre [1 : 32461], 32461 observations
- Les données de validation sont comprises entre [32462 : 32561], 100 observations.

**Test2** : Cas du sous apprentissage ; données d'entraînement (20%) supérieur aux données de validation (80%).

- Les données d'entraînement2 sont comprises entre [1 : 6512], donc 6512 observations.
- Les données de validation2 sont comprises entre [6513 : 32561], 2604 observations.

**Test3** : Cas d'apprentissage équitale; données d'entraînement (50%) supérieur aux données de validation (50%).

- Les données d'entraînement3 sont comprises entre [1 : 16280], 16280 observations
- Les données de validation3 sont comprises entre [16281 : 32561], 16281 observations.

**Test4** : données d'entraînement (60%) supérieur aux données de validation (40%).

- Les données d'entraînement4 sont comprises entre [1 : 19536]
- Les données de validation4 sont comprises entre [19537 : 32561]

## III. CHOIX DES PARAMETRES ET APPLICATION DE LA METHODE AU JEU DE DONNEES

### 1. Choix des paramètres

Le choix des attributs de notre jeu de données à appliquer à la méthode d'apprentissage supervisé est fonction de plusieurs critères.

Pour les attributs quantitatifs ou continus, nous nous sommes basé sur la corrélation entre les différentes paires. Ceci étant, si deux variables sont fortement corrélées (positivement ou négativement) nous choisissons une parmi les deux. Au cas contraire les deux sont retenues pour l'application.

Y	X	r	r <sup>2</sup>	t	Pr(> t )
Education-num	Hours-per-week	0,1481	0,0219	27,0256	0,0000
Education-num	Capital-gain	0,1226	0,0150	22,2958	0,0000
Education-num	capital-loss	0,0799	0,0064	14,4677	0,0000
Capital-gain	Hours-per-week	0,0784	0,0061	14,1918	0,0000
Age	Capital-gain	0,0777	0,0060	14,0581	0,0000
Age	Fnlwgt	-0,0766	0,0059	-13,8709	0,0000
Age	Hours-per-week	0,0688	0,0047	12,4358	0,0000
Age	capital-loss	0,0578	0,0033	10,4423	0,0000
capital-loss	Hours-per-week	0,0543	0,0029	9,8045	0,0000
Fnlwgt	Education-num	-0,0432	0,0019	-7,8014	0,0000
Age	Education-num	0,0365	0,0013	6,5954	0,0000
Capital-gain	capital-loss	-0,0316	0,0010	-5,7075	0,0000
Fnlwgt	Hours-per-week	-0,0188	0,0004	-3,3872	0,0007
Fnlwgt	capital-loss	-0,0103	0,0001	-1,8499	0,0643
Fnlwgt	Capital-gain	0,0004	0,0000	0,0779	0,9379

*Figure 1: Corrélacion entre différentes paires de variables*

D'après le rendu de la Figure 1, les attributs retenus sont :

- ✓ Education-num ;
- ✓ Hours-pers-week ;
- ✓ Capital-gain ;
- ✓ Capital-loss ;
- ✓ Age.

En ce qui concerne des attributs catégoriels, nous allons nous baser sur l'objectif de notre jeu de données pour décider sur les attributs à prédire.

- ✓ Workclass ;
- ✓ Marital-statut ;
- ✓ Occupation ;
- ✓ RelationShip ;
- ✓ Sex.

## 2. Application des paramètres à la méthode d'apprentissage supervisé

### a. Modèle du « Test1 » avec les attributs numériques

- La construction de ce modèle, se fera avec les attributs retenus après le test de corrélation des différentes paires d'attributs de la **figure 1**. L'outil utilisé dans le cadre de cette construction est bien évidemment **R**.
- La validation d'un modèle sera définie par un critère qui nous permettra de déterminer la qualité d'un modèle. L'un des plus utilisés est **Akaike Information Criterion** ou **AIC**. **Plus l'AIC sera faible, meilleur sera la modèle**. Le principe est de supprimer une variable pour chaque modèle créé et une fois constaté que la valeur de l'AIC augmente plus que la précédente alors, on n'arrête.

```
> summary(Model1Num)

Call:
glm(formula = Entrainement$Salaire ~ Entrainement$Age + Entrainement$Education.num +
    Entrainement$Hours.per.week + Entrainement$capital.gain +
    Entrainement$capital.loss, family = binomial(link = logit))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.3122  -0.6407  -0.4072  -0.1291   3.0917

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -8.324e+00  1.154e-01  -72.16  <2e-16 ***
Entrainement$Age    4.309e-02  1.224e-03   35.21  <2e-16 ***
Entrainement$Education.num  3.225e-01  6.827e-03   47.24  <2e-16 ***
Entrainement$Hours.per.week  4.094e-02  1.327e-03   30.84  <2e-16 ***
Entrainement$Capital.gain   3.185e-04  9.692e-06   32.86  <2e-16 ***
Entrainement$capital.loss   6.985e-04  3.255e-05   21.46  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 35845  on 32460  degrees of freedom
Residual deviance: 26407  on 32455  degrees of freedom
AIC: 26419

Number of Fisher Scoring iterations: 7

> |
```

Figure 2: Modèle des cinq attributs numériques

```
> summary(ModeleNum1)

Call:
glm(formula = Entrainement$Salaire ~ Entrainement$Age + Entrainement$Education.num +
    Entrainement$Hours.per.week + Entrainement$Capital.gain,
    family = binomial(link = logit))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.2330  -0.6552  -0.4174  -0.1292   3.0876

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -8.383e+00  1.145e-01  -73.20  <2e-16 ***
Entrainement$Age    4.402e-02  1.210e-03   36.37  <2e-16 ***
Entrainement$Education.num  3.297e-01  6.765e-03   48.74  <2e-16 ***
Entrainement$Hours.per.week  4.168e-02  1.316e-03   31.68  <2e-16 ***
Entrainement$Capital.gain  3.051e-04  9.572e-06   31.87  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 35845  on 32460  degrees of freedom
Residual deviance: 26871  on 32456  degrees of freedom
AIC: 26881

Number of Fisher Scoring iterations: 7
```

Figure 3: Modèle des attributs numériques avec suppression d'un

```
> ModeleNum1 <- step(ModeleNum)
Start: AIC=26419.29
Entrainement$Salaire ~ Entrainement$Age + Entrainement$Education.num +
    Entrainement$Hours.per.week + Entrainement$Capital.gain +
    Entrainement$capital.loss

              Df Deviance   AIC
<none>                26407 26419
- Entrainement$capital.loss    1    26871 26881
- Entrainement$Hours.per.week  1    27427 27437
- Entrainement$Age             1    27691 27701
- Entrainement$Capital.gain    1    28580 28590
- Entrainement$Education.num   1    29025 29035
```

Figure 4: Choix du meilleur modèle des attributs numériques

#### i. Interprétation du Modèle avec attribut numérique

Le modèle initial a un AIC de **26419.29**. A la première étape, il apparaît que la suppression de la variable **Capital-Loss** permet d'augmenter l'AIC à 26880. Alors, toute suppression de variable ferait augmenter l'AIC comme vous le constaté sur la **Figure 4**. La procédure s'arrête

et nous concluons que, le meilleur modèle est « **ModeleNum** » et justifie l'efficacité du choix des variables entrant dans la construction d'un modèle par la méthode de corrélation.

**NB** : R nous permet de sélectionner le meilleur modèle par une procédure pas à pas descendante basée sur la minimisation de l'AIC et cette fonction est « **step** ». La fonction nous affiche à l'écran les différentes étapes de la sélection et renvoi le modèle final.

## ii. Interprétation de l'Odd-ratio (Rapport de côte)

L'Odd-ratio nous permet de mesurer l'évolution du rapport de probabilités d'apparition de l'évènement  $Y=1$  contre  $Y=0$ , lorsque  $X_j$  passe de  $X_j$  à  $X_j+1$ .

- Un  $OR < 1$ , indique une influence négative de  $X_j$  sur  $Y$  ;
- Un  $OR > 1$ , indique une influence positive de  $X_j$  sur  $Y$  ;
- Un  $OR = 1$ , indique  $X_j$  est indépendant de  $Y$ .

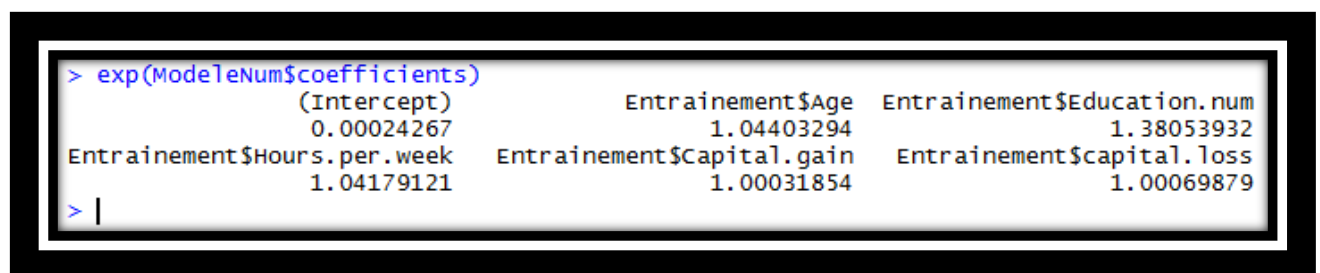


Figure 5: Odd-ratio du modèle "ModeleNum"

Nous constatons que toutes nos variables influencent positivement la variable de prédiction  $Y$  (Salaire).

## b. Modèle du « Test2 » avec les attributs numériques

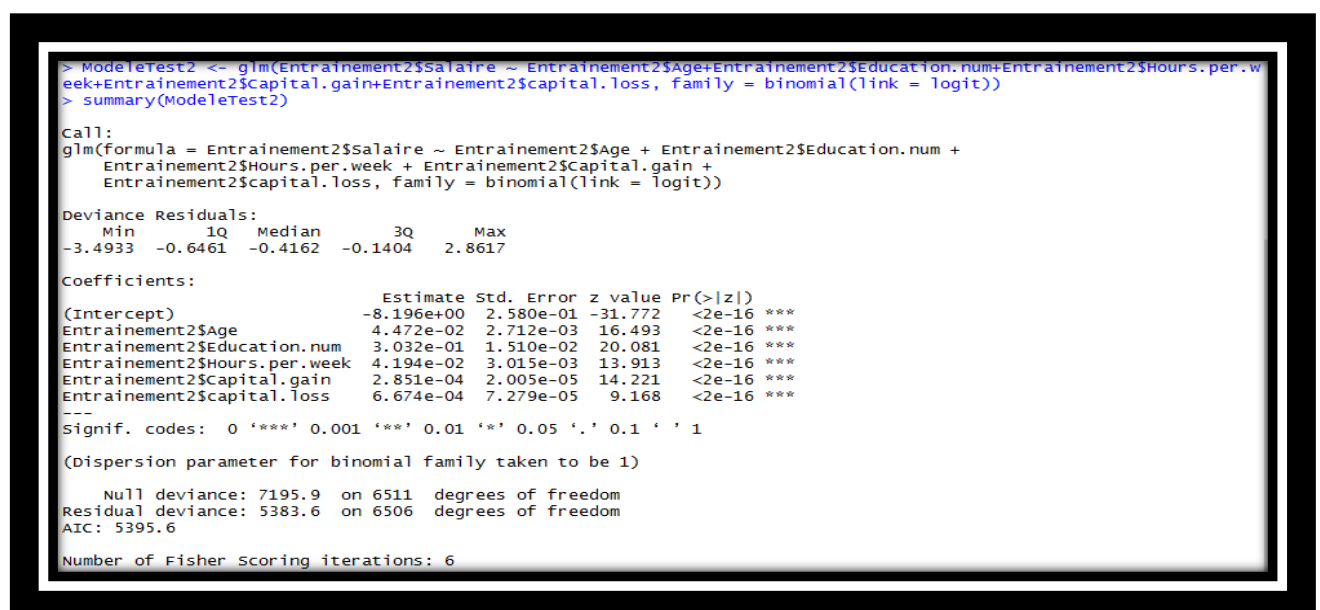


Figure 6: ModeleTest2

Supprimons un attribut « **capital-loss** » pour voir le comportement de l'AIC. Ce modèle est nommé « **ModeleTest2bis** ».

```
> ModeleTest2bis <- glm(Entrainement2$Salaire ~ Entrainement2$Age+Entrainement2$Education.num+Entrainement2$Hours.per.week+Entrainement2$Capital.gain, family = binomial(link = logit))
> summary(ModeleTest2bis)
```

Call:  
glm(formula = Entrainement2\$Salaire ~ Entrainement2\$Age + Entrainement2\$Education.num + Entrainement2\$Hours.per.week + Entrainement2\$Capital.gain, family = binomial(link = logit))

Deviance Residuals:

Min	1Q	Median	3Q	Max
-3.3322	-0.6603	-0.4246	-0.1393	2.8617

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-8.3269493	0.2568978	-32.41	<2e-16 ***
Entrainement2\$Age	0.0455515	0.0026879	16.95	<2e-16 ***
Entrainement2\$Education.num	0.3122284	0.0149704	20.86	<2e-16 ***
Entrainement2\$Hours.per.week	0.0439268	0.0029895	14.69	<2e-16 ***
Entrainement2\$Capital.gain	0.0002730	0.0000198	13.79	<2e-16 ***

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 7195.9 on 6511 degrees of freedom  
Residual deviance: 5468.3 on 6507 degrees of freedom  
AIC: 5478.3

Number of Fisher scoring iterations: 6

*Figure 7: ModeleTest2bis*

Nous constatons que la suppression d'une quelconque variable augmentera la valeur de l'AIC. Alors, le modèle « **ModeleTest2** » est le meilleur car il possède la plus petite valeur de l'AIC 5395.6. La figure ci-dessous, Figure 8, nous donne les différentes valeurs d'AIC de chaque attribut s'il venait à être supprimé.

```
> ModeleTest2bis <- step(ModeleTest2)
```

Start: AIC=5395.59

Entrainement2\$Salaire ~ Entrainement2\$Age + Entrainement2\$Education.num + Entrainement2\$Hours.per.week + Entrainement2\$Capital.gain + Entrainement2\$capital.loss

	Df	Deviance	AIC
<none>		5383.6	5395.6
- Entrainement2\$capital.loss	1	5468.3	5478.3
- Entrainement2\$Hours.per.week	1	5592.5	5602.5
- Entrainement2\$Age	1	5668.0	5678.0
- Entrainement2\$Capital.gain	1	5795.4	5805.4
- Entrainement2\$Education.num	1	5849.2	5859.2

*Figure 8: Valeurs AIC des attributs du Modèle Test2*

Nous pouvons conclure au vue des résultats que le modèle initial «**ModeleTest2** » est le meilleur avec un AIC de 5395.6 et est retenu pour effectuer notre prédiction.

### c. Modèle du « Test3 » avec les attributs numériques

```
> ModeleTest3 <- glm(Entrainement3$Salaire ~ Entrainement3$Age+Entrainement3$Education.num+Entrainement3$Hours.per.w
week+Entrainement3$Capital.gain+Entrainement3$capital.loss, family = binomial(link = logit))
warning message:
glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
> summary(ModeleTest3)

Call:
glm(formula = Entrainement3$Salaire ~ Entrainement3$Age + Entrainement3$Education.num +
    Entrainement3$Hours.per.week + Entrainement3$capital.gain +
    Entrainement3$capital.loss, family = binomial(link = logit))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.3860  -0.6428  -0.4132  -0.1401   2.9048

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -8.204e+00  1.620e-01  -50.65  <2e-16 ***
Entrainement3$Age    4.325e-02  1.717e-03   25.18  <2e-16 ***
Entrainement3$Education.num  3.152e-01  9.594e-03   32.85  <2e-16 ***
Entrainement3$Hours.per.week  3.991e-02  1.866e-03   21.39  <2e-16 ***
Entrainement3$Capital.gain  2.995e-04  1.309e-05   22.88  <2e-16 ***
Entrainement3$capital.loss  6.983e-04  4.589e-05   15.22  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 17920  on 16279  degrees of freedom
Residual deviance: 13368  on 16274  degrees of freedom
AIC: 13380

Number of Fisher Scoring iterations: 6
```

Figure 9: ModeleTest3

Supprimons un attribut « **Education-num** » pour voir le comportement de l'AIC. Ce modèle est nommé « **ModeleTest3bis** ».

```
> ModeleTest3bis <- glm(Entrainement3$Salaire ~ Entrainement3$Age+Entrainement3$Hours.per.week+Entrainement3$Capital
.gain+Entrainement3$capital.loss, family = binomial(link = logit))
warning message:
glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
> summary(ModeleTest3bis)

Call:
glm(formula = Entrainement3$Salaire ~ Entrainement3$Age + Entrainement3$Hours.per.week +
    Entrainement3$Capital.gain + Entrainement3$capital.loss,
    family = binomial(link = logit))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.6006  -0.6822  -0.5034  -0.2357   2.6778

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -4.893e+00  1.107e-01  -44.21  <2e-16 ***
Entrainement3$Age    3.898e-02  1.577e-03   24.71  <2e-16 ***
Entrainement3$Hours.per.week  4.401e-02  1.785e-03   24.65  <2e-16 ***
Entrainement3$Capital.gain  3.136e-04  1.228e-05   25.53  <2e-16 ***
Entrainement3$capital.loss  7.777e-04  4.289e-05   18.13  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

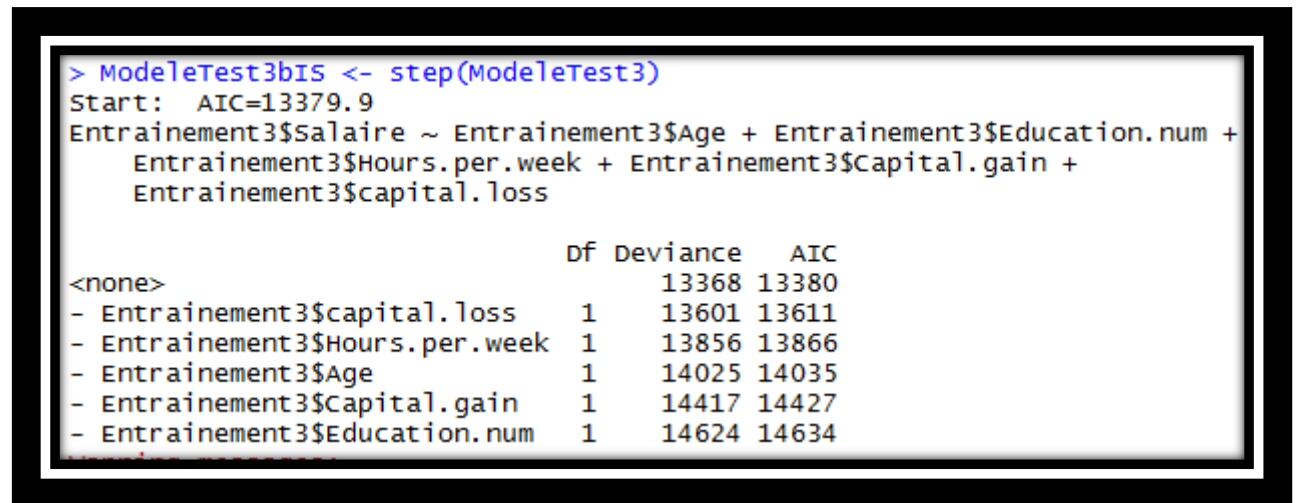
    Null deviance: 17920  on 16279  degrees of freedom
Residual deviance: 14624  on 16275  degrees of freedom
AIC: 14634

Number of Fisher Scoring iterations: 6
```

Figure 10: ModeleTest3bis



Nous constatons que la suppression d'une quelconque variable augmentera la valeur de l'AIC. Alors, le modèle « **ModeleTest3** » est le meilleur car il possède la plus petite valeur de l'AIC 13380. La figure ci-dessous, Figure 11, nous donne les différentes valeurs d'AIC de chaque attribut s'il venait à être supprimé.



*Figure 11: Valeurs AIC des attributs du Modèle Test3*

Nous pouvons conclure au vue des résultats que le modèle initial «**ModeleTest3** » est le meilleur avec un AIC de **13380** et est retenu pour effectuer notre prédiction.

En définitive, la construction des trois modèles **ModeleNum**, **ModeleTest2** et **ModeleTest3** qui nous permettrons de prédire ont été validé par la valeur de l'AIC que nous avons donné le principe à la section III. 2. a . La prédiction sera faite à partir des trois modèles. Ceci pour avoir plus de clarté sur les probabilités ou les pourcentages de prédiction qui, est de déterminer **si une personne fait plus de 50k par année.**

#### **d. Modèle du « Test4 » avec les attributs numériques**



```

> Entrainement4 <- data.frame(Adult_Data[1:19536,])
> view(Entrainement4)
> validation4 <- data.frame(Adult_Data[19537:32561,])
> view(validation4)
> ModeleTest4 <- glm(Entrainement4$Salaire ~ Entrainement4$Age+Entrainement4$Education.num+Entrainement4$Hours.per.w
week+Entrainement4$Capital.gain+Entrainement4$capital.loss, family = binomial(link = logit))
warning message:
glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
> summary(ModeleTest4)

Call:
glm(formula = Entrainement4$Salaire ~ Entrainement4$Age + Entrainement4$Education.num +
    Entrainement4$Hours.per.week + Entrainement4$Capital.gain +
    Entrainement4$capital.loss, family = binomial(link = logit))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.4372  -0.6453  -0.4137  -0.1432   2.9142

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -8.191e+00  1.478e-01  -55.41  <2e-16 ***
Entrainement4$Age    4.250e-02  1.566e-03   27.14  <2e-16 ***
Entrainement4$Education.num  3.174e-01  8.787e-03   36.12  <2e-16 ***
Entrainement4$Hours.per.week  3.954e-02  1.708e-03   23.14  <2e-16 ***
Entrainement4$Capital.gain  3.071e-04  1.227e-05   25.02  <2e-16 ***
Entrainement4$capital.loss  7.214e-04  4.183e-05   17.25  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 21473  on 19535  degrees of freedom
Residual deviance: 15984  on 19530  degrees of freedom
AIC: 15996

Number of Fisher Scoring iterations: 7

```

*Figure 12: Modele "Test4"*

Supprimons un attribut « **Education-num** » pour voir le comportement de l'AIC. Ce modèle est nommé « **ModeleTest4bis** ».

La suppression des attributs dans les modèles précédents augmentait la valeur de l'AIC. Par conséquent, nous pouvons considérer ce modèle « Test4 » comme meilleur. Alors, il sera utilisé pour effectuer la prédiction.

#### e. Modèle avec les attributs catégoriels

- La construction de ce modèle, se fera avec les attributs retenus intuitivement par nous en fonction de l'objectif de notre jeu de données. L'outil utilisé dans le cadre de cette construction est bien évidemment **R**.
- La validation d'un modèle sera définie par un critère qui nous permettra de déterminer la qualité d'un modèle. L'un des plus utilisés est **Akaike Information Criterion** ou **AIC**. **Plus l'AIC sera faible, meilleur sera le modèle**. Le principe est de supprimer une variable pour chaque modèle créé et une fois constaté que la valeur de l'AIC augmente plus que la précédente alors, on n'arrête.

```

> ModeleDis<-glm(Entrainement$Salaire~Entrainement$workclass+Entrainement$Education+Entrainement$Marital.statut+E
ntrainement$Occupation+Entrainement$Relationship+Entrainement$Sex+Entrainement$Race+Entrainement$Native.country,
family = binomial(link = logit))
> summary(ModeleDis)

Call:
glm(formula = Entrainement$Salaire ~ Entrainement$workclass +
    Entrainement$Education + Entrainement$Marital.statut + Entrainement$Occupation +
    Entrainement$Relationship + Entrainement$Sex + Entrainement$Race +
    Entrainement$Native.country, family = binomial(link = logit))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4231  -0.5813  -0.2264  -0.0421   3.6867

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    -3.348926   0.705700  -4.746 2.08e-06 ***
Entrainement$workclassLocal-gov    -0.605394   0.103443  -5.852 4.84e-09 ***
Entrainement$workclassNever-worked -12.084625  276.711423  -0.044 0.965166
Entrainement$workclassPrivate      -0.491339   0.085356  -5.756 8.60e-09 ***
Entrainement$workclassSelf-emp-inc  -0.012969   0.111587   0.116 0.907474
Entrainement$workclassSelf-emp-not-inc -0.760505   0.099513  -7.642 2.13e-14 ***
Entrainement$workclassState-gov     -0.890407   0.114839  -7.754 8.94e-15 ***
Entrainement$workclassWithout-pay   -13.256466  195.071907  -0.068 0.945820
Entrainement$Education11th         -0.019727   0.198625  -0.099 0.920887
Entrainement$Education12th          0.440392   0.243861   1.806 0.070932 .
Entrainement$Education1st-4th       -0.544072   0.460571  -1.181 0.237484
Entrainement$Education5th-6th       -0.199011   0.307060  -0.648 0.516909
Entrainement$Education7th-8th       -0.415274   0.219685  -1.890 0.058716 .
Entrainement$Education9th           -0.316030   0.249471  -1.267 0.205227
Entrainement$EducationAssoc-acdm     1.217356   0.164179   7.415 1.22e-13 ***
Entrainement$EducationAssoc-voc      1.249318   0.157496   7.932 2.15e-15 ***
Entrainement$EducationBachelors      1.874221   0.146274  12.813 < 2e-16 ***
Entrainement$EducationDoctorate      3.270257   0.201015  16.269 < 2e-16 ***
Entrainement$EducationHS-grad        0.724857   0.143017   5.068 4.01e-07 ***
Entrainement$EducationMasters        2.391503   0.156197  15.311 < 2e-16 ***
Entrainement$EducationPreschool     -11.106789  105.656062  -0.105 0.916279

Entrainement$Native.countryJamaica   -1.563100   0.726523  -2.151 0.031438 *
Entrainement$Native.countryJapan     -0.901374   0.685659  -1.315 0.188641
Entrainement$Native.countryLaos      -1.760322   0.918577  -1.916 0.055320 .
Entrainement$Native.countryMexico    -1.974791   0.626472  -3.152 0.001620 **
Entrainement$Native.countryNicaragua -2.401900   1.010566  -2.377 0.017464 *
Entrainement$Native.countryOutlying-US(Guam-USVI-etc) -13.953842  206.777730  -0.067 0.946198
Entrainement$Native.countryPeru       -2.673162   0.999278  -2.675 0.007471 **
Entrainement$Native.countryPhilippines -0.894277   0.615350  -1.453 0.146145
Entrainement$Native.countryPoland     -1.313337   0.702719  -1.869 0.061632 .
Entrainement$Native.countryPortugal  -1.702502   0.856861  -1.987 0.046933 *
Entrainement$Native.countryPuerto-Rico -1.638983   0.693034  -2.365 0.018033 *
Entrainement$Native.countryScotland  -1.287857   0.954569  -1.349 0.177289
Entrainement$Native.countrySouth      -2.023018   0.677304  -2.987 0.002819 **
Entrainement$Native.countryTaiwan     -1.695949   0.706599  -2.400 0.016388 *
Entrainement$Native.countryThailand   -1.787290   1.003791  -1.781 0.074988 .
Entrainement$Native.countryTrinidad&Tobago -2.317731   1.095265  -2.116 0.034333 *
Entrainement$Native.countryUnited-States -1.151742   0.598746  -1.924 0.054406 .
Entrainement$Native.countryVietnam    -2.405172   0.797242  -3.017 0.002554 **
Entrainement$Native.countryYugoslavia -0.682087   0.873011  -0.781 0.434624
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 35845  on 32460  degrees of freedom
Residual deviance: 23344  on 32369  degrees of freedom
AIC: 23528

Number of Fisher Scoring iterations: 13

```

*Figure 13: Modèle des attributs catégoriels*

```

> ModeleDis1<-step(ModeleDis)
Start: AIC=23528.31
Entrainement$Salaire ~ Entrainement$workclass + Entrainement$Education +
  Entrainement$Marital.statut + Entrainement$occupation + Entrainement$Relationship +
  Entrainement$Sex + Entrainement$Race + Entrainement$Native.country

              Df Deviance   AIC
<none>                23344 23528
- Entrainement$Race      4    23361 23537
- Entrainement$Native.country 40    23450 23554
- Entrainement$workclass   7    23489 23659
- Entrainement$Sex         1    23521 23703
- Entrainement$Marital.statut 6    23548 23720
- Entrainement$Relationship 5    23725 23899
- Entrainement$occupation 13    24083 24241
- Entrainement$Education 15    24779 24933
> |

```

*Figure 14: Choix du meilleur modèle des attributs discrets*

Le modèle initial a un AIC de **23528**. A la première étape, il apparaît que la suppression de l'attribut **Race** à la construction du modèle permettra d'augmenter l'AIC à **23537** Idem pour les autres attributs. Alors, toute suppression d'attribut ferait augmenter l'AIC comme vous le constaté sur la **Figure 7**. La procédure s'arrête et nous concluons que, le meilleur modèle est « **ModeleDis** »

### 3. Prédiction : Evaluation de la capacité d'un modèle

Dans les étapes précédentes, nous avons évalué de manière brève l'ajustement du modèle, nous allons maintenant voir comment le modèle se comporte lors de la prédiction de Y sur un nouveau jeu de données. Cette prédiction se fera aussi avec R, en définissant le type de paramètre = « réponse », R nous ressort des probabilités sous la forme **P (Y=1| X)**

#### a. Prédiction du modèle « ModeleNum » des attributs numériques

**Rappel : Test1** : Cas du sur apprentissage ; données d'entraînement (99%) supérieur aux données de validation (1%).

- Les données d'entraînement sont comprises entre [1 : 32461], 32461 observations
- Les données de validation sont comprises entre [32462 : 32561], 100 observations.

```

> Entrainement=subset(Validation,select = c(1,5,11,12,13))
> View(Entrainement)
> Probability=predict(ModeleNum, newdata = Entrainement,type = "reponse")
Error in match.arg(type) :
  'arg' should be one of "link", "response", "terms"
> Probability=predict(ModeleNum, newdata = Entrainement,type = "response")
> Probability
 32462 32463 32464 32465 32466 32467 32468 32469 32470
0.14340054 0.73463551 0.21676858 0.41333956 0.17451034 0.99129474 0.08500290 0.25354595 0.96733602
 32471 32472 32473 32474 32475 32476 32477 32478 32479
0.73834974 0.13640925 0.47694424 0.51871033 0.31564048 0.06715926 0.31072372 0.11974033 0.12880752
 32480 32481 32482 32483 32484 32485 32486 32487 32488
0.06670616 0.40016578 0.28942143 0.11081032 0.05156016 0.19892383 0.07061181 0.06711172 0.39056688
 32489 32490 32491 32492 32493 32494 32495 32496 32497
0.11081032 0.23951784 0.06338646 0.08612659 0.17298198 0.03246334 0.14613403 0.21123481 0.01129519
 32498 32499 32500 32501 32502 32503 32504 32505 32506
0.05678182 0.20954190 0.06258490 0.17796753 0.11770850 0.35976611 0.14696585 0.59687486 0.03457970
 32507 32508 32509 32510 32511 32512 32513 32514 32515
0.29165761 0.41594812 0.05244663 0.09972739 0.13026475 0.19517861 0.05107389 0.28368267 0.22959837
 32516 32517 32518 32519 32520 32521 32522 32523 32524
0.30964008 0.07576967 0.02093918 1.00000000 0.24018167 0.16276150 0.08612659 0.12680080 0.13273110
 32525 32526 32527 32528 32529 32530 32531 32532 32533
0.06182509 0.22351900 0.03316965 0.07492403 0.07958083 0.06071399 0.40813714 0.77105910 0.68078196
 32534 32535 32536 32537 32538 32539 32540 32541 32542
0.56029278 0.12920829 0.03347018 0.39777066 0.09573853 0.98420208 0.57558104 0.13648678 0.08748973
 32543 32544 32545 32546 32547 32548 32549 32550 32551
0.21493237 0.36590470 0.22354946 0.12403788 0.22756938 0.12664397 0.89250611 0.16679847 0.23164002
 32552 32553 32554 32555 32556 32557 32558 32559 32560
0.03316965 0.25325870 0.12132965 0.52805004 0.07492403 0.14996437 0.11302211 0.21676858 0.02521676
 32561
0.96238085
> |

```

Figure 15: Probabilités des attributs numériques du modèle "ModeleNum"

### Interpretation de la figure 14.

Dans la **figure 14**, nous avons deux éléments importants de notre analyse à savoir, le numéro d'une observation surligné en jaune et en dessous sa probabilité.

Ceci étant, pour chaque observation, nous avons la probabilité de celle-ci

```

> fitted.Probability<-ifelse(Probability > 0.5,1,0)
> View(fitted.Probability)
> fitted.Probability
32462 32463 32464 32465 32466 32467 32468 32469 32470 32471 32472 32473 32474 32475 32476 32477 32478 32479
0 1 0 0 0 1 0 0 1 1 0 0 1 0 0 0 0 0
32480 32481 32482 32483 32484 32485 32486 32487 32488 32489 32490 32491 32492 32493 32494 32495 32496 32497
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
32498 32499 32500 32501 32502 32503 32504 32505 32506 32507 32508 32509 32510 32511 32512 32513 32514 32515
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
32516 32517 32518 32519 32520 32521 32522 32523 32524 32525 32526 32527 32528 32529 32530 32531 32532 32533
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1
32534 32535 32536 32537 32538 32539 32540 32541 32542 32543 32544 32545 32546 32547 32548 32549 32550 32551
1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0
32552 32553 32554 32555 32556 32557 32558 32559 32560 32561
0 0 0 1 0 0 0 0 0 1
> |

```

Figure 16: Probabilité du modèle "ModeleNum" selon le seuil défini à 0.5

### Interprétation de Figure 15

- « 0 » appartient à la classe des personnes faisant un revenu annuel de <=50k
- « 1 » appartient à la classe des personnes faisant un revenu annuel de plus de >50k
- Le chiffre au-dessus de « 0 » ou « 1 » est une observation.

## Matrice de confusion

Cette matrice nous permet de donner le **taux d'erreur de prédiction du modèle**.

```
> table(fitted.Probability,Revenu=validation$Salaire)
      Revenu
fitted.Probability 0  1
0      74 11
1       5 10
> |
```

Figure 17: Matrice de confusion du modèle "ModeleNum"

Dans notre cas, nous avons un **taux d'erreur de 16%**, équivalent au calcul de **5+11=16**.

```
> misClasificError <- mean(fitted.Probability != validation$Salaire)
> print(paste('Accuracy',1-misClasificError))
[1] "Accuracy 0.84"
> print(misClasificError)
[1] 0.16
> |
```

Figure 18: Estimation des taux de bonne classification et de mauvaise classification du « ModeleNum »

Nous pouvons conclure que la précision de **0.84** sur l'ensemble de validation est un bon résultat.

### b. Prédiction du modèle « ModeleTest2 » des attributs numériques

**Rappel : Test2** : Cas du sous apprentissage ; données d'entraînement (20%) supérieur aux données de validation (80%).

- Les données d'entraînement2 sont comprises entre [1 : 6512], donc 6512 observations.
- Les données de validation2 sont comprises entre [6513 : 32561], 2604 observations.

**NB : Les résultats seront présentés par les matrices de confusion**

```
> table(fitted.ProbabilityTest2, RevenuTest2=validation2$Salaire)
      RevenuTest2
fitted.ProbabilityTest2 0    1
0      18754  3803
1       1025  2467
> |
```

Figure 19: Matrice de confusion du modèle "ModeleTest2"

Le taux d'erreur dans ce modèle est de **18.5%**

```

> misClasificError <- mean(fitted.ProbabilityTest2 != validation2$Salaire)
> print(paste('Accuracy', 1-misClasificError))
[1] "Accuracy 0.814656992590886"
> print(misClasificError)
[1] 0.185343
> |

```

*Figure 20: Estimation de taux de bonne classification et de mauvaise pour le "ModeleTest2"*

Nous pouvons conclure que la précision de **81.5%** sur l'ensemble de validation2 est un bon résultat.

### c. Prédiction du modèle « ModeleTest3 » des attributs numériques

**Rappel : Test3 :** Cas d'apprentissage équitable; données d'entraînement (**50%**) supérieur aux données de validation (**50%**).

- Les données d'entraînement3 sont comprises entre [1 : 16280], 16280 observations
- Les données de validation3 sont comprises entre [16281 : 32561], 16281 observations.

```

> table(fitted.ProbabilityTest3, RevenuTest3=validation3$Salaire)
      RevenuTest3
fitted.ProbabilityTest3    0    1
0      11733   2387
1       604   1557
> |

```

*Figure 21: Matrice de confusion du modèle "ModeleTest3"*

Le taux d'erreur dans ce modèle est de **18.37%**

```

> misClasificError <- mean(fitted.ProbabilityTest3 != validation3$Salaire)
> print(paste('Accuracy', 1-misClasificError))
[1] "Accuracy 0.816288925741662"
> print(misClasificError)
[1] 0.1837111
> |

```

*Figure 22: Estimation du taux de bonne classification et de mauvaise du modèle "ModeleTest3"*

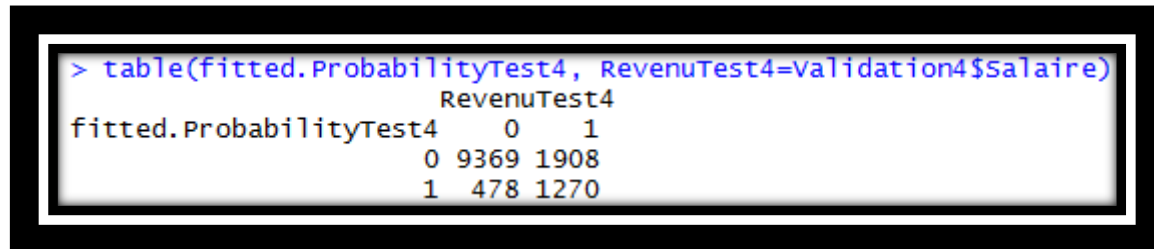
Nous pouvons conclure que la précision de **81.63%** sur l'ensemble de validation3 est un bon résultat.



#### d. Prédiction du modèle « ModeleTest3 » des attributs numériques

**Rappel : Test4** : données d'entraînement (60%) supérieur aux données de validation (40%).

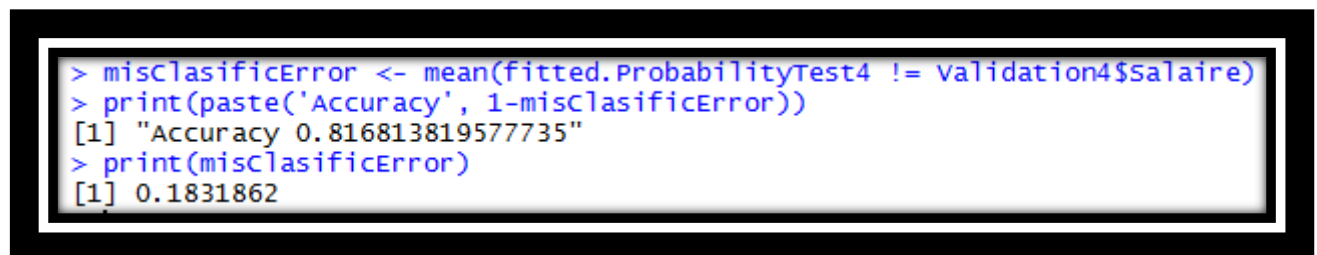
- Les données d'entraînement4 sont comprises entre [1 : 19536]
- Les données de validation4 sont comprises entre [19537 : 32561]



```
> table(fitted.ProbabilityTest4, RevenuTest4=validation4$Salaire)
      RevenuTest4
fitted.ProbabilityTest4  0    1
0      9369 1908
1      478 1270
```

*Figure 23: Matrice de confusion "Modele Test4"*

Le taux d'erreur dans ce modèle est de **18.32%**



```
> misClasificError <- mean(fitted.ProbabilityTest4 != validation4$Salaire)
> print(paste('Accuracy', 1-misClasificError))
[1] "Accuracy 0.816813819577735"
> print(misClasificError)
[1] 0.1831862
```

*Figure 24: Estimation du taux de bonne classification et de mauvaise du modèle « ModeleTest4 »*

Nous pouvons conclure que la précision de **81.68%** sur l'ensemble de validation4 est un bon résultat

## IV. CLASSIFICATION PAR ARBRE DE DECISION

On donne un ensemble  $X$  de  $N$  exemples notés  $X_i$  donc les Attributs sont quantitatifs et/ou qualitatifs. Chaque exemple  $X$  est étiqueté, c'est-à-dire qu'il lui est associé « une classe » ou « un attribut cible » que l'on note  $Y$ . De par cet exemple, nous construisons un arbre dit « de décision » tel que :

- Chaque nœud correspond à un test sur la valeur d'un ou plusieurs attributs ;
- Chaque branche partant d'un nœud corresponde à une ou plusieurs valeurs de ce test.

Un arbre de décision peut être exploité de plusieurs manières :

- En  $Y$  classant de nouvelle données ;
- En faisant de l'estimation d'attribut ;
- En extrayant un jeu de règle de classification concernant l'attribut cible ;
- En interprétant la pertinence des attributs.

# 1. Construction d'un arbre de décision des attributs numériques

## a. Test1 : Construction du modèle « Modele\_Arbre »

```
> Modele_Arbre
n= 32461

node), split, n, loss, yval, (yprob)
* denotes terminal node

1) root 32461 7820 <=50K (0.75909553 0.24090447)
2) Relationship=Not-in-family,other-relative,Own-child,Unmarried 17745 1175 <=50K (0.93378416 0.06621584)
4) Capital.gain< 7073.5 17428 870 <=50K (0.95008033 0.04991967) *
5) Capital.gain>=7073.5 317 12 >50K (0.03785489 0.96214511) *
3) Relationship=Husband,wife 14716 6645 <=50K (0.54845067 0.45154933)
6) Education.num< 12.5 10296 3447 <=50K (0.66520979 0.33479021)
12) Capital.gain< 5095.5 9777 2938 <=50K (0.69949882 0.30050118) *
13) Capital.gain>=5095.5 519 10 >50K (0.01926782 0.98073218) *
7) Education.num>=12.5 4420 1222 >50K (0.27647059 0.72352941) *
```

Figure 25: Modèle pour la construction de l'arbre

Ce dernier nous donne un ensemble des règles textuelles pour la classification de nouvelles données.

## b. Prédiction et interprétation des résultats

```
> t(Prediction_Arbre)
32462 32463 32464 32465 32466 32467 32468 32469 32470
<=50K 0.95012012 0.01915709 0.6998063 0.2763989 0.95012012 0.2763989 0.6998063 0.95012012 0.95012012
>50K 0.04987988 0.98084291 0.3001937 0.7236011 0.04987988 0.7236011 0.3001937 0.04987988 0.04987988
32471 32472 32473 32474 32475 32476 32477 32478 32479
<=50K 0.95012012 0.95012012 0.2763989 0.95012012 0.6998063 0.95012012 0.6998063 0.95012012 0.95012012
>50K 0.04987988 0.04987988 0.7236011 0.04987988 0.3001937 0.04987988 0.3001937 0.04987988 0.04987988
32480 32481 32482 32483 32484 32485 32486 32487 32488
<=50K 0.6998063 0.2763989 0.95012012 0.6998063 0.95012012 0.95012012 0.6998063 0.95012012 0.6998063
>50K 0.3001937 0.7236011 0.04987988 0.3001937 0.04987988 0.04987988 0.3001937 0.04987988 0.3001937
32489 32490 32491 32492 32493 32494 32495 32496 32497
<=50K 0.95012012 0.6998063 0.95012012 0.95012012 0.95012012 0.95012012 0.95012012 0.95012012 0.95012012
>50K 0.04987988 0.3001937 0.04987988 0.04987988 0.04987988 0.04987988 0.04987988 0.04987988 0.04987988
32498 32499 32500 32501 32502 32503 32504 32505 32506
<=50K 0.95012012 0.6998063 0.95012012 0.95012012 0.6998063 0.2763989 0.6998063 0.95012012 0.95012012
>50K 0.04987988 0.3001937 0.04987988 0.04987988 0.3001937 0.7236011 0.3001937 0.04987988 0.04987988
32507 32508 32509 32510 32511 32512 32513 32514 32515 32516
<=50K 0.6998063 0.2763989 0.95012012 0.95012012 0.6998063 0.95012012 0.95012012 0.2763989 0.6998063 0.6998063
>50K 0.3001937 0.7236011 0.04987988 0.04987988 0.3001937 0.04987988 0.04987988 0.7236011 0.3001937 0.3001937
32517 32518 32519 32520 32521 32522 32523 32524 32525
<=50K 0.95012012 0.6998063 0.01915709 0.6998063 0.95012012 0.6998063 0.6998063 0.95012012 0.95012012
>50K 0.04987988 0.3001937 0.98084291 0.3001937 0.04987988 0.3001937 0.3001937 0.04987988 0.04987988
32526 32527 32528 32529 32530 32531 32532 32533 32534
<=50K 0.95012012 0.6998063 0.95012012 0.6998063 0.95012012 0.2763989 0.95012012 0.2763989 0.2763989
>50K 0.04987988 0.3001937 0.04987988 0.3001937 0.04987988 0.7236011 0.04987988 0.7236011 0.7236011
32535 32536 32537 32538 32539 32540 32541 32542 32543
<=50K 0.95012012 0.95012012 0.95012012 0.95012012 0.03773585 0.2763989 0.95012012 0.95012012 0.6998063
>50K 0.04987988 0.04987988 0.04987988 0.04987988 0.96226415 0.7236011 0.04987988 0.04987988 0.3001937
32544 32545 32546 32547 32548 32549 32550 32551 32552
<=50K 0.95012012 0.95012012 0.6998063 0.95012012 0.6998063 0.95012012 0.95012012 0.6998063 0.6998063
>50K 0.04987988 0.04987988 0.3001937 0.04987988 0.3001937 0.04987988 0.04987988 0.3001937 0.3001937
32553 32554 32555 32556 32557 32558 32559 32560 32561
<=50K 0.6998063 0.95012012 0.2763989 0.95012012 0.6998063 0.6998063 0.95012012 0.95012012 0.01915709
>50K 0.3001937 0.04987988 0.7236011 0.04987988 0.3001937 0.3001937 0.04987988 0.04987988 0.98084291
> |
```

Figure 26: Prédiction du modèle "Modele\_Arbre"

- Le premier numéro (exemple 32462) surligné en jaune correspond au numéro de l'observation de nos données de validation.



- Le deuxième est sa probabilité d'appartenir à la classe  $\leq 50$
- Le troisième est sa probabilité d'appartenir à la classe  $>50k$

Ceci est valable pour toutes les observations du jeu de données.

#### c. Matrice de confusion du test1

```
> table(Revenu_Predire, Revenu_Reel=Validation_Arbre$Prediction)
      Revenu_Reel
Revenu_Predire <=50K >50K
      <=50K      76      8
      >50K       3     13
> |
```

Figure 27: Matrice de confusion du modèle "Modele\_Arbre"

Dans notre cas, nous avons un **taux d'erreur de 11%**.

#### d. Matrice de confusion du test2

```
> table(Revenu_Predire, Revenu_Reel=Validation_Arbre2$Prediction)
      Revenu_Reel
Revenu_Predire <=50K >50K
      <=50K 18794  3028
      >50K   985  3242
> |
```

Figure 28: Matrice de confusion du modèle "Modele\_Arbre2"

Taux d'erreur=15.41% Taux de précision=84,59%

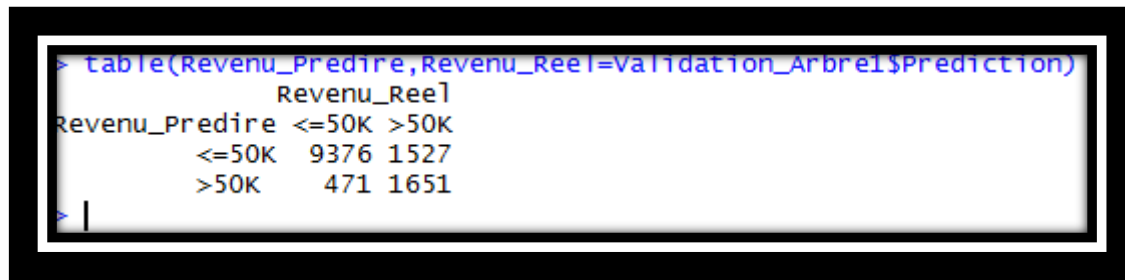
#### e. Matrice de confusion du test3

```
> table(Revenu_Predire, Revenu_Reel=Validation_Arbre3$Prediction)
      Revenu_Reel
Revenu_Predire <=50K >50K
      <=50K 11730  1891
      >50K   607  2053
> |
```

Figure 29: Matrice de confusion du modèle "Modele\_Arbre3"

Taux d'erreur=15.34% Taux de précision=84,66%

#### f. Matrice de confusion du test4



```
> table(Revenu_Predire, Revenu_Reel=Validation_Arbre4$Prediction)
      Revenu_Reel
Revenu_Predire <=50K >50K
      <=50K    9376 1527
      >50K     471 1651
```

Figure 30: Matrice de confusion du modèle "Modele\_Arbre4"

Taux d'erreur=15.34% Taux de précision=84,66%

## V. COMPARAISON DES RESULTATS DES DEUX METHODES D'APPRENTISSAGE SUPERVISEES

La comparaison des deux méthodes que nous avons utilisées se focalisera sur deux points : La complexité et puis l'estimation du taux d'erreur.

Du point de vue complexité, le temps de calcul est plus coûteux avec l'arbre de décision car, ce dernier cherche à chaque étape de la construction du classifieur le meilleur nœud parmi les 14 variables explicatives.

Du point de vue d'estimation du taux d'erreur, nous avons un taux de la régression logistique binaire et du classificateur d'arbre de décision.

	Régression binaire	Arbre de décision (ID3)
Test1	Taux d'erreur= 16% Taux de pression=84%	Taux d'erreur= 11% Taux de pression=89%
Test2	Taux d'erreur= 18.5% Taux de pression=81.5%	Taux d'erreur=15.41% Taux de précision=84,59%
Test3	Taux d'erreur= 18.37% Taux de pression=81.63%	Taux d'erreur=15.34% Taux de précision=84,66%
Test4	Taux d'erreur= 18.32% Taux de pression=81.68%	Taux d'erreur=15.34% Taux de précision=84,66%

Nous remarquons surtout dans le cas des tests (test2, test3, test4) le modèle avec l'arbre de décision à un taux d'erreur inférieur à celui du modèle de régression.

## CONCLUSION

Dans ce travail, nous avons proposé une méthode d'apprentissage supervisée (**Régression Logistique Binaire**) appliquée à un jeu de donnée « Adult ». L'objectif de la classification supervisée étant de définir les règles permettant de classer les objets dans des classes à partir des variables quantitatives ou qualitatives caractérisant ces objectifs. Les méthodes s'étendent souvent à des variables quantitatives. Il est nécessaire d'étudier la fiabilité de ces règles pour les comparer et les appliquer, évaluer les cas de sous apprentissage et de sur apprentissage (complexité du modèle). Pour le faire, nous utilisons deuxième échantillon indépendant, dit de validation ou de test.

## Références

- Régression logistique.htm
- Fouille de données Notes de cours Ph. PREUX Université de Lille 3  
[philippe.preux@univ-lille3.fr](mailto:philippe.preux@univ-lille3.fr) 26 mai 2011. Pages
- <https://www.r-bloggers.com/how-to-perform-a-logistic-regression-in-r/>
- [https://www.tutorialspoint.com/r/r\\_decision\\_tree.htm](https://www.tutorialspoint.com/r/r_decision_tree.htm)
- Mathilde Mougeot Université Paris-Diderot - Paris 7
- [mathilde.mougeot@univ-parisdiderot.fr](mailto:mathilde.mougeot@univ-parisdiderot.fr) pages 35
- La régression logistique  
Par Sonia NEJI et Anne-Hélène JIGOREL. Page 11.
- Fouille de données  
Cours 4 - Exploration des données multidimensionnelles - Apprentissage supervisé  
NGUYỄN Thị Minh Huyền c 2016
- Classification supervisée. Aperçu de quelques méthodes avec le logiciel R