Université d'Evry Val d'Essonne

Master II: Innovation, Marché et Sciences des Données
(MII IMSD)

Evaluation Finale du Scoring

Rapport du Projet

Jiajun MA
Chandanika UN

Année Scolaire : 2019-2020

Table de matières

Description **2**

Partie A **2**

Partie B **9**

Index de figure: **10**

Index de tableau: **10**

Annexe: **10**

Description

Dans la pratique, c'est souvent qu'on a besoins de prédire la variable binomial. Par exemple, dans la banque on a besoins une modèle pour prédire si la client est une bonne client ou pas.

Dans notre projet, on modélise le défaut d'un portefeuille composé de plusieurs entreprises afin d'anticiper le comportement d'entités nouvelles. On a 5 variables quantitatives:

- WCTA : Le fonds de roulement divisé par le total actifs de l'entreprise
- RETA : Le bénéfice non distribué rapporté au total actifs
- EBIT_TA : Le revenu brut d'exploitation
- METL : La valeur du marché des actifs divisé par le total passif
- STA : Total du ventes

Et notre variable cible binomial est:

- Défaut : Elle vaut 0 si l'entreprise a eu un défaut de paiement et 1 sinon.

Partie A

1. Les différents modèles (paramétriques et non paramétriques y compris les arbres de décisions) adaptés à ce jeu de données.

- Modèle paramétrique:
  - Logistic Regression
  - GMM (Gaussian Mixture Model)
- Modèle non paramétrique:
  - Decision Tree
  - SVM
  - K-Means
  - Random Forest
  - Adaboosting

2. On décide de retenir un modèle de régression logistique. Les 3 écritures rigoureuses de ce modèle.

Le modèle de régression logistique: On veut expliquer le variable binaire Y par p variables explicatives $X_1, \ldots, X_p$.

1. Présentation de modèle [Hosmer and Lemeshow, 2000]:

Les observations y_i sont des réalisations de variables aléatoire Y_i indépendantes de loi Bernoulli de paramètre $p_\beta(x_i)$ tel que:

$$logit\, p_\beta(x_i) = log(\frac{p_\beta(x_i)}{1-p_\beta(x_i)}) = \beta_1 x_{i1} + \ldots + \beta_p x_{ip} = x'_{i\beta}$$

2. Fonction odd:

$odds = \frac{P}{1-P}$ où P *est la probabilité qu'un événement est réalisé.*

$logit(P) = log(odds) = ln(\frac{P}{1-P}) = \beta_0 + \beta x$

$\Rightarrow (\frac{P}{1-P}) = exp^{(\beta_0+\beta x)}$

$$\Rightarrow P = \frac{exp^{(\beta_0+\beta x)}}{1+exp^{(\beta_0+\beta x)}}$$

3. On suppose qu'on a une seule variable explicative X, et suppose qu'il existe une seule variable latente (inobservéé) Y^* :

$Y^* = \tilde{\beta}_0 + \beta_1 x_i + \varepsilon$ où ε *est une variable centrée, telle que* $Y_i = 1_{Y^*_i > s}$, $s \in \mathbb{R}$
On a alors : $P(Y_i = 1) = P(-\varepsilon < \beta_0 + \beta_1 x_i) = F_\varepsilon(\beta_0 + \beta_1 x_i)$ ou $\beta_0 = \tilde{\beta}_0 - s$.
Si ε suit une loi logistique , c'est à dire de fonction de répartition:
$F_\varepsilon(x) = \frac{exp(x)}{1+exp(x)}$ alors le modèle est le modèle logistique.

1) Gradient Descent Method

$$P(y;\ x,\ \beta) = \frac{1}{1+e^{(-(x\beta+\beta_0))}}$$
$$\Rightarrow P(\ y = 1|x,\ \beta) = h_\beta(x)$$
$$P(y = 0|x,\ \beta) = 1 - h_\beta(x)$$
$$donc,\ P(y|x,\ \beta) = h^{\ y}{}_\beta(x).(1 - h_\beta(x))^{(1-y)}$$
$$max\ L(\theta) = \prod_{i=1}^{m} h_\beta(x^{(i)})^{y(i)}(1 - h_\beta(x))^{1-y(i)}$$
$$min - l(\theta) = -ln(L(\theta)) = -\sum_{i=1}^{m}[y_{(i)} ln h_\beta(x_i) + (1 - y_i) ln(1 - h_\beta(x_i))]$$
$$\therefore \frac{\delta(l(\theta))}{\delta\beta_j} = -\sum_{i=1}^{m}(y_i - h_\beta(x_i)x^{(i)}{}_j$$
$$\therefore \theta_{j+1} = \theta_j + a\sum_{i=1}^{m}(y^{(i)} - h_\beta(x^{(i)}))x^{(i)}{}_j$$

2) Newton's Method

$$l'(\theta + \Delta\theta) = l(\theta) + l'(\theta)\Delta\theta + \tfrac{1}{2}l''(\theta)\Delta\theta^2$$
$$quand\ \Delta\theta \to 0$$
$$l'(\theta) + \tfrac{1}{2}l''(\theta)\Delta\theta = 0$$
$$\Rightarrow l'(\theta) + l''(\theta)\Delta\theta = 0$$
$$\Delta\theta = -\frac{l'(\theta)}{l''(\theta)}$$
$$\therefore \theta_{j+1} = \theta_j - \frac{l'(\theta)}{l''(\theta)}$$
$$avec\ l'(\theta) = \sum_{i=1}^{m} x^{(i)}{}_j(y^{(i)} - h_\theta(x^{(i)}))$$
$$l''(\theta) = \sum_{i=1}^{m} x^{(i)}{}_j h_\theta(x^{(i)})[h_\theta(x^{(i)}) - 1]x^{(i)}{}_k$$

3) IRLS

$$\beta_{n+1} = (XAX^T + \lambda I)^{-1}XAZ$$
$$o\grave{u}\ Z = X^T\beta_n + t$$
$$t_i = \frac{y_i[1-h_\beta(y_i w^T x_i)]}{A_{ii}}$$

$$A = \sum_{i=1}^{n} \hat{y}_i(1 - \hat{y}_i)$$

3. La méthode qu'on utilise pour estimer le modèle précédent est :

- La méthode du gradient descent.

4. l'algorithme IRLS :

L'algorithme IRLS qui s'appelle aussi la méthode de Newton-Raphson, est une méthode itérative, telle que X_{t+1} est fonction de X_t , qui vise à trouver l'argmax ou l'argmin d'un fonction f. On fait un développement de Taylor autour de X_t , ce qui :

$$f(x) = f(x_t) + \frac{f}{x}(x_t)(x - x_t) + \frac{1}{2}(x - x_t)^T \frac{\delta^2 f}{xx^T}(x_t)(x - x_t) + (o\|x - x_t\|^2)$$

Où $f(x_t) + \frac{f}{x}(x_t)(x - x_t) + \frac{1}{2}(x - x_t)^T \frac{\delta^2 f}{xx^T}(x_t)(x - x_t) = \hat{f}_t(x)$

Le principe de la méthode de Newton-Raphson est de déterminer x_{t+1} à partir de $\hat{f}_t$.

Plus précisément on définit x_{t+1} comme suit : $x_{t+1} = min\,(f_t(x))$.

$$\frac{\hat{f}}{x} = \frac{f}{x}(x_t) + \frac{\delta^2 f}{x\delta x^T}(x_t)(x - x_t) = 0$$
$$\Leftrightarrow -(\frac{\delta^2 f}{xx^T}(x_t))^{-1} \frac{f}{x}(x_t) = x - x_t$$
$$\Leftrightarrow x = x_t - (\frac{\delta^2 f}{\delta x \delta x^T}(x_t))^{-1} \frac{\delta f}{\delta x}(x_t)$$

5. Lors de votre estimation, un collègue (non statisticien) vous propose d'utiliser la méthode des moindres carrés ordinaires pour estimer ce modèle, notre réaction (arguments à l'appui) :

MCO n'est pas une bonne choix pour résoudre notre problème car:

- Le variable cible Y dans le modèle de logistique régression suivent la distribution binomiale et la MCO est normalement utilisé dans les cas qui suivent la loi normale.
- Si on utilise la méthode des moindres carrés, la fonction loss de logistique régression qui n'est pas convexe, elle n'est donc pas facile à résoudre et il tombe facilement dans un optimum local.
- Pour le modèle de logistique régression, on veut trouver la probabilité de Y égale 1 ou 0 qui est entre 0 et 1, mais si on utilise la MCO pour estimer ce modèle, il nous rend probablement le résultat numérique continue qui peut être négative ou supérieur à 1, ce qui n'est pas logique.

6. Les valeurs manquantes, aberrantes et extrêmes dans les données:

- Les valeurs manquantes : le graphique au dessous nous montre qu'il existe pas de valeur manquante dans cette base de données

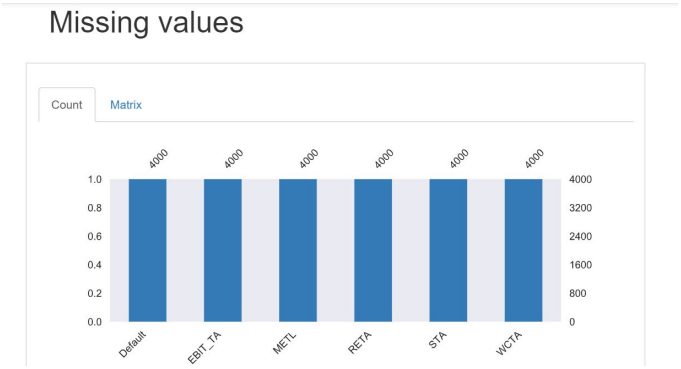


Figure 1 : Nombre de valeur manquante dans les données

- Les valeurs aberrantes et extrêmes :

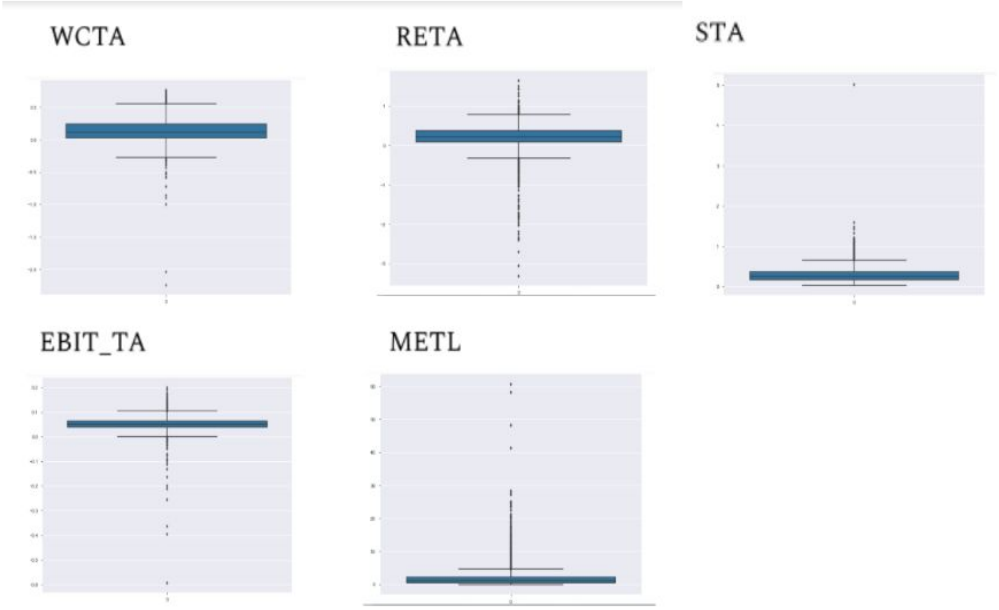


Figure 2 : Box Plots des variables quantitatives

En regardant les boxplot au dessus, on peut constater les petits points noirs à l'extérieur de l'intervalle entre les deux lignes, qui sont des points aberrants et extrême des variables.

7. Une analyse descriptive des variables y compris la variable défaut (statistiques descriptives des variables, croisement entre les variables explicatives, croisement entre la variable à expliquer et les variables explicatives avec les indicateurs statistiques appropriés).

a. Etude univariée

- EBIT_TA

Quantile statistics

Minimum	-0.5918213
5-th percentile	0.017640795
Q1	0.03899205
Median	0.05156275
Q3	0.06509305
95-th percentile	0.092121665
Maximum	0.1980351
Range	0.7898564
Interquartile range	0.026101

Descriptive statistics

Standard deviation	0.02959551396
Coef of variation	0.5713063527
Kurtosis	85.99742655
Mean	0.05180322925
MAD	0.01819561136
Skewness	-4.844361643
Sum	207.212917
Variance	0.0008758944467
Memory size	31.3 KiB

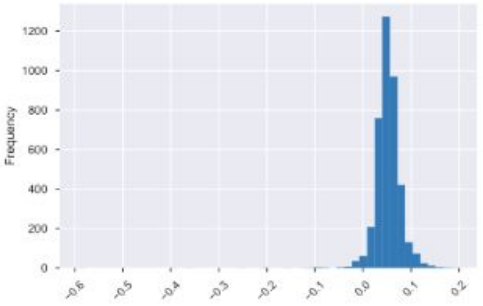


Figure 3 : Analyse Descriptive : EBIT_TA

- METL

Quantile statistics

Minimum	0.0237178
5-th percentile	0.220168505
Q1	0.6234926
Median	1.1364555
Q3	2.24428725
95-th percentile	5.5976636
Maximum	60.60715
Range	60.5834322
Interquartile range	1.62079465

Descriptive statistics

Standard deviation	2.994474998
Coef of variation	1.532513892
Kurtosis	103.1275367
Mean	1.953962711
MAD	1.512719264
Skewness	7.750728118
Sum	7815.850843
Variance	8.966880513
Memory size	31.3 KiB

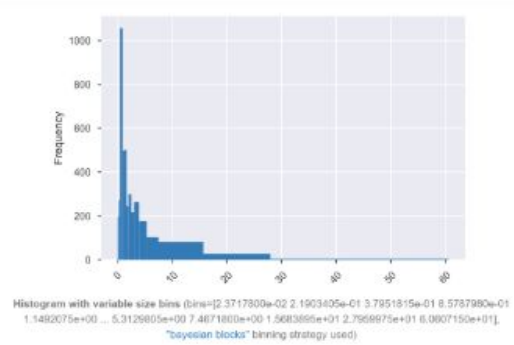


Figure 4 : Analyse Descriptive : METL

- RETA

Quantile statistics

Minimum	-3.31242
5-th percentile	-0.25125144
Q1	0.094744125
Median	0.21892865
Q3	0.3734173
95-th percentile	0.65049419
Maximum	1.639646
Range	4.952066
Interquartile range	0.278673175

Descriptive statistics

Standard deviation	0.3325494467
Coef of variation	1.580054402
Kurtosis	17.44119599
Mean	0.2104670866
MAD	0.2081209656
Skewness	-2.554891512
Sum	841.8683465
Variance	0.1105891345
Memory size	31.3 KiB

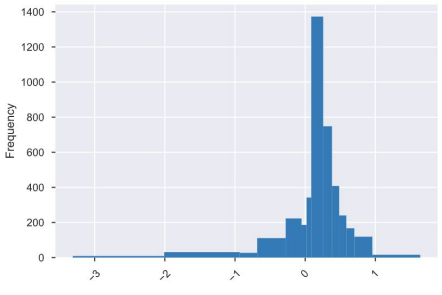


Figure 5 : Analyse descriptive : RETA

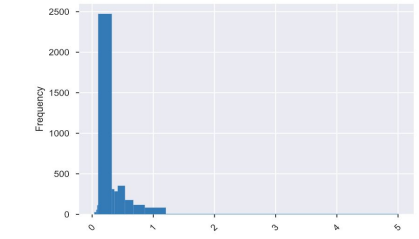
- STA

Quantile statistics

Minimum	0.0358583
5-th percentile	0.09858544
Q1	0.170835675
Median	0.26103665
Q3	0.36693605
95-th percentile	0.680471925
Maximum	5.007775
Range	4.9719167
Interquartile range	0.196100375

Descriptive statistics

Standard deviation	0.2057940454
Coef of variation	0.6777304598
Kurtosis	71.21524143
Mean	0.3036517578
MAD	0.1384536794
Skewness	4.48100441
Sum	1214.607031
Variance	0.04235118914
Memory size	31.3 KiB



Histogram with variable size bins (bins=[0.0358583 0.0676626 0.08299195 0.0983963 0.32168285 ... 0.537605 0.6732202 0.86149175 1.204981 5.007775], "bayesian blocks" binning strategy used)

Figure 6 : Analyse Descriptive : STA

- WCTA

Quantile statistics

Minimum	-2.240268
5-th percentile	-0.06212881
Q1	0.031166775
Median	0.1172729
Q3	0.24177525
95-th percentile	0.44222302
Maximum	0.7660402
Range	3.0063082
Interquartile range	0.210608475

Descriptive statistics

Standard deviation	0.1707626102
Coef of variation	1.198219109
Kurtosis	17.6822686
Mean	0.1425136762
MAD	0.1275981464
Skewness	-1.014531708
Sum	570.054705
Variance	0.02915986903
Memory size	31.3 KiB

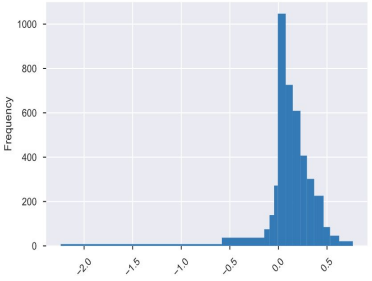


Figure 7 : Analyse Descriptive : WCTA

- En regardant les graphiques au dessus, nous pouvons avoir des données statistiques descriptives pour chaque variables et nous pouvons constater visuellement que des variables tels que METL, STA et WCTA ne sont pas normales.

Fréquence: Variable qualitatif: *y*

Default
Boolean

Distinct count	2
Unique (%)	< 0.1%
Missing (%)	0.0%
Missing (n)	0

Toggle details

Value	Count	Frequency (%)
1	3928	98.2%
0	72	1.8%

Figure 8 : Analyse Descriptive de variable DEFAULT

Ici, la variable DEFAULT contient des fréquence fiable de 0 comparer à la fréquence de valeur 1. On utilise ces données tandis le petit nombre de 0, car notre nombre d'observation total est grande.

b. Etude multivariée

- Entre les variables quantitatives: (Coefficient de correlation)

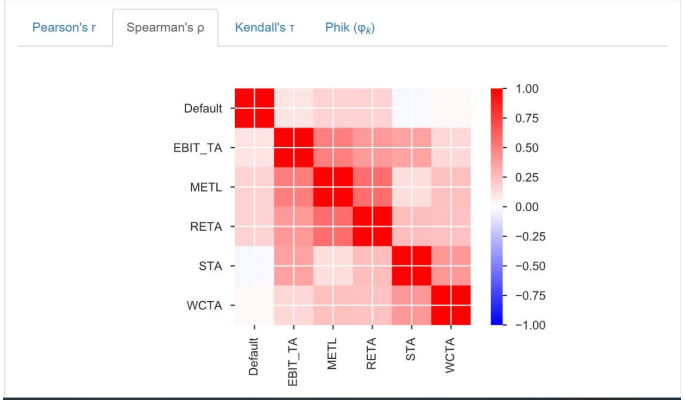


Figure 9 : Matrice de correlation

- On utilise d'ici le test non-paramétrique de Spearman parce que nos variables ne sont pas normales. On peut donc voir qu'ils se sont corrélés.

8. Les conséquences de la multicolinéarité :

- Une multicolinéarité peut augmenter la variance des coefficients de régression et les rendre instables et difficiles à interpréter
- Si la multicolinéarité est parfaite alors la matrice $(X^T X)^{-1}$ n'est pas inversible, alors l'estimateur MCO n'est pas calculable
- Lorsque l'une des variables explicatives est proche d'une combinaison linéaire, des autres variables alors $(X^T X)$ serait mal conditionnée ($det(X^T X)$ proche de 0). $(X^T X)^{-1}$ aura des éléments très grands
- Lorsque des termes d'un modèle sont fortement corrélés, la suppression de l'un de ces termes aura une incidence considérable sur les coefficients estimés des autres. Les coefficients des termes fortement corrélés peuvent même présenter le mauvais signe.

9. Implémentation des modèles de régression logistique :

- $y = f(WCTA, RETA)$
- $y = f(METL, EBIT_{TA})$
- $y = f(WCTA, RETA, EBIT_{TA}, STA)$
- $y = f(WCTA, RETA, EBIT_{TA}, METL, STA)$

Interprétation pour chaque modèle les coefficients estimés, et les courbes ROC associées à chaque modèle ainsi les aires sous ces courbes.

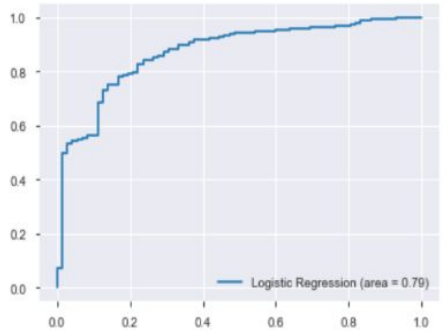
1) $y = f(WCTA, RETA)$

Coefficients estimés:

WCTA: -0.12119685

RETA: 1.59170473

Courbe ROC:



Aire sous ROC:

ROC_AUC: 0.7902481330617787

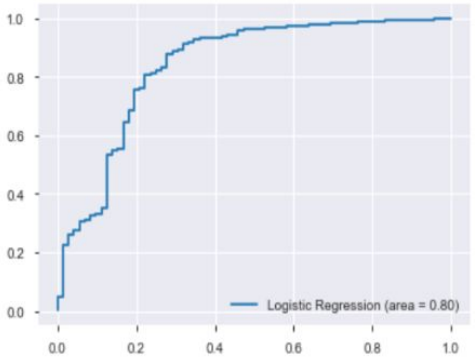
2) $y = f(METL, EBIT\_TA)$

Coefficients estimés:

METL: 0.45176068

EBIT_TA: 2.21847612

Courbe ROC:



Aire sous ROC:

ROC_AUC: 0.7966310251188051

3) $y = f(WCTA, RETA, EBIT\_TA, STA)$

Coefficients estimés:

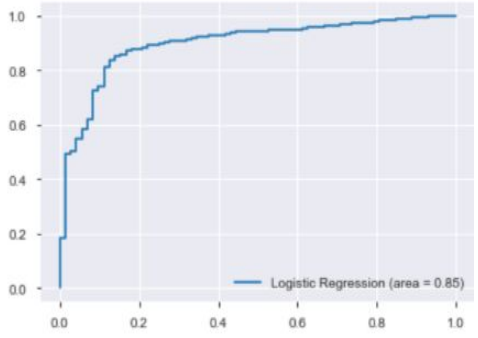
WCTA: -0.07852652

RETA: 1.46347653

EBIT_TA: 0.69099715

STA: -0.56467775

Courbe ROC:



Aire sous ROC:

ROC_AUC: 0.8526391717583164

4) $y = f(WCTA, RETA, EBIT\_TA, METL, STA)$

Coefficients estimés:

WCTA: -0.11068791

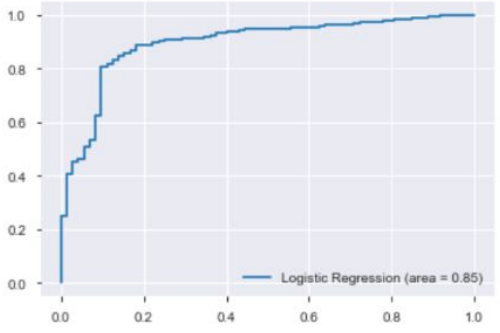
RETA: 1.34993391

EBIT_TA: 0.59047754

METL: 0.511959

STA: -0.51424021

Courbe ROC:



Aire sous ROC:

ROC_AUC: 0.8509136682507354

Figure 10 : Différents modèles avec ses coefficients et courbe ROC

- Selon le figure 10, on a vu que notre modèle amélioré en comparer l'aire sous la courbe de ROC. Cette amélioration continue jusqu'à le 3ème modèle. On peut remarquer qu'ajouter un variable de plus dans le modèle créer un pire modèle.

10. La modèle choisir selon les critères du BIC/SC et de l'AIC.

Modèle	AIC/BIC
$y = f(WCTA, RETA)$	AIC: [4742.86890792] BIC: [4760.68103201]
$y = f(METL, EBIT\_TA)$	AIC: [5402.69773574] BIC: [5420.50985983]
$y = f(WCTA, RETA, EBIT\_TA, \quad STA)$	AIC: [5022.20387492] BIC: [5051.8907484]
$y = f(WCTA, RETA, EBIT\_TA, METL, STA)$	AIC: [5139.53214978] BIC: [5175.15639795]

Table 1 : AIC/BIC pour chaque modèle (Partie A)

En comparant les AIC et BIC de chaque modèle, on a décidé de choisir le premier Modèle avec que des variable WCTA et RETA qui en a le minimum AIC et BIC.

La définition de ces critères AIC et BIC/SC et leur logique de construction:

Le critère d'information d'Akaike(le critère d'information bayésien) est une mesure de la qualité d'un modèle statistique. Lorsque l'on estime un modèle statistique, il est possible d'augmenter la vraisemblance du modèle en ajoutant un paramètre. Ces critères permettent de pénaliser les modèles en fonction du nombre de paramètres afin de satisfaire le critère de parcimonie. On choisit alors le modèle avec le AIC ou le BIC le plus faible.

$$AIC = 2 * k - 2 * ln(L)$$

$$BIC/SC = k * ln(N) - 2 * ln(L)$$

L :La vraisemblance du modèle estimée, N: Le nombre d'observations dans l'échantillon, et k: Le nombre de paramètres libres du modèle

11. On décide de retenir le modèle avec 5 variables. La probabilité de défaut d'une nouvelle entreprise ayant les caractéristiques suivants :

(WCTA = 0.6, RETA= 0.25, EBIT_TA= 0.45, METL= 0.05, STA= 0.72s)

- Si on retient le modèle avec 5 variables, les coefficients estimés sont [-0.17952814 0.66707738 0.2975832 2.25491858 -0.15455624], et on peut calculer la probabilité de défaut de cette entreprise étant de 45.15%.

Partie B

Nous décidons d'implémenter le même modèle non plus sur les variables quantitatives directement mais dorénavant sur des variables discrétisées.

1. Les avantages de ce approche

- 1) L'augmentation et la diminution des caractéristiques discrètes sont faciles et il est facile d'itérer rapidement le modèle.

- 2) L'opération de multiplication de produits internes à vecteur clairsemé est rapide, les résultats de calcul sont pratiques à stocker et faciles à développer.

- 3) Les entités discrétisées sont très robustes aux données anormales: par exemple, une entité dont l'âge est supérieur à 30 vaut 1, sinon 0. Si les caractéristiques ne sont pas discrétisées, une donnée anormale "300 ans" causera de grandes interférences avec le modèle.

- 4) La régression logistique est un modèle linéaire généralisé avec une puissance expressive limitée. Une fois la variable unique discrétisée en N, chaque variable a un poids distinct, ce qui équivaut à introduire une non-linéarité dans le modèle, ce qui peut améliorer la puissance expressive du modèle et augmenter l'ajustement ;

- 5) Après discrétisation, un croisement de caractéristiques peut être effectué, des variables M + N aux variables M * N, introduisant davantage la non-linéarité et améliorant la capacité d'expression;

- 6) Une fois les fonctionnalités discrétisées, le modèle sera plus stable. Par exemple, si l'âge de l'utilisateur est discrétisé, 20-30 comme intervalle ne deviendra pas une personne complètement différente car un utilisateur a un an de plus. Bien sûr, les échantillons adjacents à l'intervalle seront exactement le contraire, alors comment diviser l'intervalle est très important.;

- 7) Après discrétisation des caractéristiques, il simplifie le modèle de régression logistique et réduit le risque de sur-ajustement du modèle

2. On est obligés de poser des contraintes d'identifiabilité avant l'estimation, car:

- Le principe de l'hypothèse du méthode discrétisation des valeurs continues est que des différents intervalles de valeurs continues ont des contributions différentes aux résultats.

3. Une discrétisation des 5 variables quantitatives avec les contraintes suivantes : 2 classes au moins et 3 classes au plus pour chaque variable.

- On a appliqué la méthode de K Means pour les rassembler en 3 clusters et puis on calcule la moyenne de chaque centre de cluster en tant que le point de split.

- WCTA
  Class1 : x <= 0.10 ; Classe2 : 0.10 < x <= 0.30 ; Classe3 : x > 0.30
- RETA
  Class1 : x <= -0.44 ; Classe2 : -0.44 < x <= 0.29 ; Classe3 : x > 0.29
- EBIT_TA
  Class1 : x <= -0.14 ; Classe2 : -0.14 < x <= 0.05 ; Classe3 : x > 0.05
- METL
  Class1 : x <= 2.74 ; Classe2 : 2.74 < x <= 11.51 ; Classe3 : x > 11.51
- STA
  Class1 : x <= 0.28 ; Classe2 : 0.28 < x <= 0.61 ; Classe3 : x > 0.61

	WCTA_0	WCTA_1	WCTA_2	RETA_0	RETA_1	RETA_2	EBIT_TA_0	EBIT_TA_1	EBIT_TA_2	METL_0	METL_1	METL_2	STA_0	STA_1	STA_2
0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	0.0
1	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0
2	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	0.0
3	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	0.0
4	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	0.0
5	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	0.0

Figure 11 : Les données discrétisées

4. La corrélation entre les différentes variables discrétisées à partir d'indicateurs adaptés:

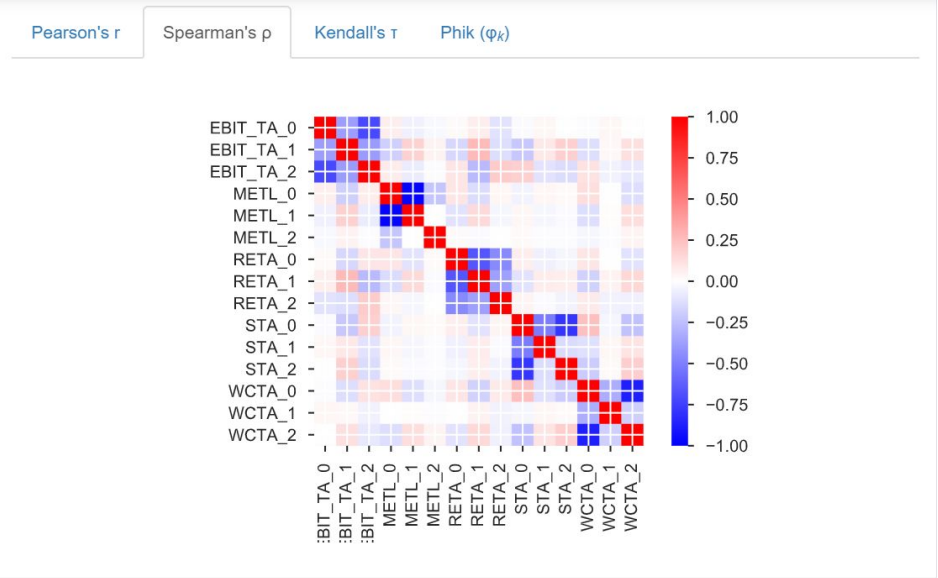
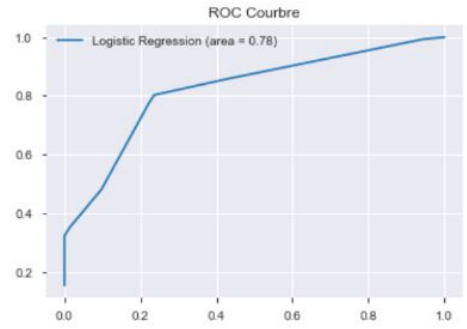


Figure 12 : Matrice de corélation

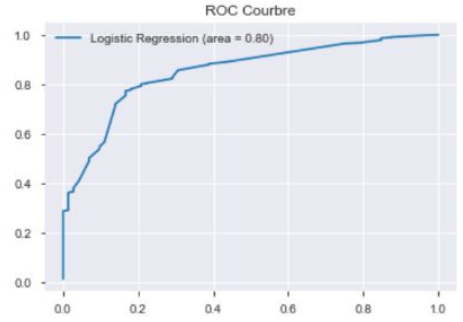
- En regardant ce graphique de corrélation entre des variables discrétisées, on peut trouver qu'elles se sont moins corrélées entre eux par rapport à celles avant de faire la discrétisation, ce qui nous permet d'entraîner un modèle plus pertinent au cas normal ayant moins de risques de overfitting.

5. Estimer un modèle de régression logistique sur les variables quantitatives discrétisées avec une méthode pas à pas. Interprétez les paramètres et les différents indicateurs de performance (AIC, BIC/SC, courbe ROC, indice de Gini,…).

$y = f(WCTA, RETA)$
AIC : [3793.29029681]
BIC : [3828.91454499]
Coubre :



$y = f(WCTA, RETA, EBIT\_TA, STA)$
AIC : [6598.15460022]
BIC : [6669.40309657]
Courbe ROC :



$y = f(WCTA, RETA, EBIT\_TA, METL, STA)$
AIC : [3688.28386788]
BIC : [3723.90811606]:
Courbe ROC :

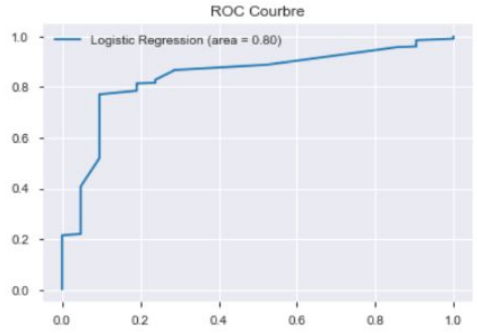


Figure 13 : Les coefficients et Courbe ROC

On a vu que notre modèle amélioré quand on ajoute les variables dans le modèle.

6. Comparez les différents modèles estimés lors de la question précédente et sélectionnez le meilleur modèle selon le critère de notre choix.

- En comparant des critères tels que le AIC, le BIC et le score de ROC_AUC, on a décidé de sélectionner le modèle avec des 5 variables : *WCTA*,*RETA*, *EBIT_TA*, *METL*, *STA* qui a le AIC et BIC les plus faibles et le ROC AUC le plus grand.

7. Implémenter une grille de score sur une échelle de 1000.

Notre formule pour calculer le score:

$c^j = \frac{max\{x^j{}_1,\dots,x^j{}_p;0\} - min\{x^j{}_1,\dots,x^j{}_p;0\}}{\sum_{j=1}^{k} max\{x^j{}_1,\dots,x^j{}_p;0\} - min\{x^j{}_1,\dots,x^j{}_p;0\}}$ *ou* $x^j{}_1, \dots, x^j{}_p$ *sont des coefficients de variable quantitatif* j*, qui est discrétisé en p variables.*

Variables	Score d'echelle de 1000
WCTA	135
RETA	405
EBIT_TA	74
METL	284
STA	102

Table 2 : Grille de Score en échelle de 1000

Index de figure

Figure 1 : Nombre de valeur manquante dans les données
Figure 2 : Box Plots des variables quantitatives
Figure 3 : Analyse Descriptive : EBIT_TA
Figure 4 : Analyse Descriptive : METL
Figure 5 : Analyse descriptive : RETA
Figure 6 : Analyse Descriptive : STA
Figure 7 : Analyse Descriptive : WCTA
Figure 8 : Analyse Descriptive de variable DEFAULT
Figure 9 : Matrice de correlation
Figure 10 : Différents modèles avec ses coefficients et courbe ROC

Index de tableau

Tableau 1 : AIC/BIC pour chaque modèle (Partie A)
Table 2 : Grille de Score en échelle de 1000

Annexe

Notre code en Python:

```
from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix,roc_curve, classification_report
from sklearn.metrics import classification_report
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
from math import log
from sklearn.cluster import KMeans
from sklearn.preprocessing import OneHotEncoder

def test_statistique():
    data = impute_data()
    print(" ")
    for column in data.columns[1:]:
        # print des boxplot pour chaque column(variable) pour identifier visuallement des points extremes
        print( column + "'s boxplot")
        f,ax=plt.subplots(figsize=(10,8))
        sns.boxplot(data=data['STA'],ax=ax)
        plt.show()

print("================================================================================
============================")
        print(" "

def impute_data():
    data = pd.read_excel("C:\\Users\\57621\\Desktop\\IMSD\\scoring\\projet\\data_scoring.xlsx")
    data = data.iloc[:,2:]
    return data

def preprocessing_data(data,columns_list):
```

```
    data_x = pd.DataFrame(data,columns=columns_list)
    scaler = StandardScaler()
    # centrer et réduire des données
    x = scaler.fit_transform(data_x)
    y = np.array(data.iloc[:,0],dtype=int)
    return x,y

def discretisation_data(x,columns_list):
    x = pd.DataFrame(x,columns=columns_list)
    # Utiliser la méthode KMeans pour rassembler des individues en 3 clusters
    km = KMeans(n_clusters=3,n_jobs=4)
    for column in columns_list:
        km.fit(np.array(data[column]).reshape(-1,1))
        c = pd.DataFrame(km.cluster_centers_).sort_values(0)
        w = []
        for i in [0,1]:
            # Trouver des points de splits en utilisant la moyenne entre deux centre de cluster
            mean = (c.iloc[i+1]+c.iloc[i])/2
            w.append(mean)

        # Spliter la column en 3 classes et leur attribuer une valuer différente
        if column == "WCTA" or column == "STA":
            x[column].loc[x[column] <= list(w[0])[0]] = 0
            x[column].loc[x[column] > list(w[1])[0]] = 2
            x[column].loc[(x[column] <= list(w[1])[0])&(x[column] > list(w[0])[0])] = 1
        elif column == "RETA" or column == "EBIT_TA":
            x[column].loc[x[column] > list(w[1])[0]] = 1
            x[column].loc[(x[column] <= list(w[1])[0])&(x[column] > list(w[0])[0])] = 0
            x[column].loc[x[column] <= list(w[0])[0]] = 2
        else:
            x[column].loc[x[column] <= list(w[0])[0]] = 0
            x[column].loc[x[column] > list(w[1])[0]] = 20
            x[column].loc[(x[column] <= list(w[1])[0])&(x[column] > list(w[0])[0])] = 10
    new_columns_list = []
    for i in range(len(columns_list)):
        new_columns_list.append(columns_list[i] + "_0")
        new_columns_list.append(columns_list[i] + "_1")
        new_columns_list.append(columns_list[i] + "_2")

    # Utiliser la méthode de Onehot pour discréstiser la variable
    one = OneHotEncoder()
    new_x = pd.DataFrame(one.fit_transform(x).toarray(),columns=new_columns_list)

    return new_x


def split_data(x,y):
    # Split la data en datat_train et data_test
```

```
    x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state=22)
    return x_train,x_test,y_train,y_test

def search_parameters(x_train,y_train):
    # Trouver des parametres du modèle les plus performantes en utilisant GridSearchCV
    k = int(sum(y_train==1)/sum(y_train==0))
    class_weight = []
    for i in range(k-5,k+5):
        class_weight.append({1:1,0:i})
    parameters = {'C':[0.01,0.1,1,5,10,100],'class_weight':class_weight}
    lg = LogisticRegression()
    lg_clf = GridSearchCV(lg,parameters,cv=5,n_jobs=3,verbose=5,scoring='roc_auc')
    lg_clf.fit(x_train,y_train)
    return lg_clf.best_params_

def train(columns_list,discre=False):

    for i in columns_list:
        if i not in ['Default', 'WCTA', 'RETA', 'EBIT_TA', 'METL', 'STA']:
            print("varaible inconnu")
            return 0
    data = impute_data()
    x,y = preprocessing_data(data,columns_list)
    # Décide si nous ferrons la dicrétisation de données ou pas
    if discre:
        x = discretisation_data(x,columns_list)
    x_train,x_test,y_train,y_test = split_data(x,y)
    best_param = search_parameters(x_train,y_train)
    lg =
LogisticRegression(C=best_param["C"],class_weight=best_param["class_weight"],solver='lbfgs',m
ax_iter=1000,random_state=22)
    lg.fit(x_train,y_train)
    return lg,x_train,x_test,y_train,y_test

def model_result(columns_list,discre=False):
    lg,x_train,x_test,y_train,y_test = train(columns_list,discre=discre)
    y_pred = lg.predict(x_test)
    y_pred_proba = lg.predict_proba(x_test)[:,1]
    print("Coefficient du modele")
    print(lg.coef_)
    print(" ")
    print("==========================================================================")
    print("")
    print("AIC and BIC")
    w = lg.coef_
    sse = 0
    for i in range(len(x_train)):
        k = x_train[i].dot(w.T)
```

```
        sse += k*y_train[i]-log(np.exp(k)+1)
    aic = 2 * len(data.columns) - 2*sse
    bic = log(len(y_train)) * len(data.columns) - 2*sse
    print("aic: "+str(aic))
    print("bic: "+str(bic))
    print("======================================================================")
    print(" ")
    print("Classification Report")
    print(classification_report(y_test,y_pred))
    print(" ")
    print(" ======================================================================")

    confusion =confusion_matrix(y_test,y_pred)
    indices = range(len(confusion))
    plt.figure()
    plt.xticks(indices, [0,1])
    plt.yticks(indices, [0,1])
    for first_index in range(len(confusion)):
        for second_index in range(len(confusion[first_index])):
            plt.text(first_index, second_index, confusion[first_index][second_index])
    plt.imshow(confusion.T, cmap=plt.cm.Blues)
    plt.ylabel('guess')
    plt.xlabel('fact')
    plt.title("Confusion Matrix")
    plt.colorbar()

    plt.figure()
    logit_roc_auc = roc_auc_score(y_test, y_pred)
    fpr, tpr, thresholds = roc_curve(y_test, y_pred_proba)
    plt.plot(fpr, tpr, label='Logistic Regression (area = %0.2f)' % logit_roc_auc)
    plt.title("ROC Courbre")
    plt.legend()

def grill_score(df):
    s = 0.0
    d = []
    for i in range(len(df)//3):
        diff = df.iloc[i*3:(i+1)*3].max() - df.iloc[i*3:(i+1)*3].min()
        s += diff
    for j in range(len(df)//3):
        diff = df.iloc[i*3:(i+1)*3].max() - df.iloc[i*3:(i+1)*3].min()
        d.append(round((diff/s)*1000)
    return d
```