

MRSQ_2015 > transformation devis en société(s) ?

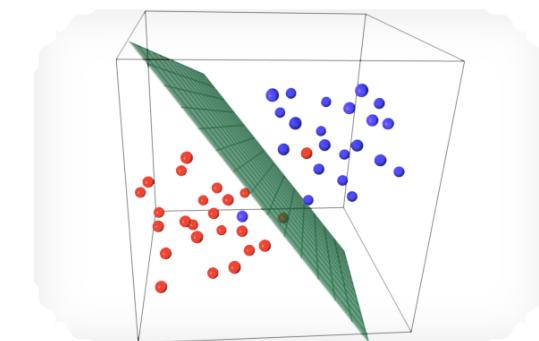


Auteur : Didier LEPICAUT
Période : Période 4
Insa - Jury : THESE PROFESSIONNELLE - INSA - Bruno PORTIER
Objet : Contrat de Professionnalisation à la MATMUT/SCD - Valérie BRIVET
Date : 27/10/2017
Insa - Jury : Mr Jean MAQUET - Directeur des formations et de la vie étudiante
Matmut : Me Stéphanie Ginnestet, Mr Johann Pelfrene

Problème : le rôle des Machines à Vecteur de Support pour prédire la transformation d'un devis commercial en contrat d'assurance automobile ?

CADRAGE

| | |
|--------------------------------|---|
| date | 27.octobre.2017 |
| version | version 3.2 |
| source des données | données MRSQ - année 2015 |
| nombre de variables | 112 « features » - variables |
| nombre d'enregistrements | 728674 lignes |
| périmètre devis concerné | Devis auto (hors 2 roues) et particulier (hors usage professionnel) |
| problème | classification binaire supervisée avec CLASSES DESEQUILIBREES |
| type d'analyse | modèle de discrimination avec sélection de variables quantitatives |
| Note destiné au lecteur | Présentations des résultats clefs |



- SOMMAIRE -

| | |
|----------------------------------|-------|
| (1) Contexte | p. 3 |
| (2) Problème | p. 4 |
| (3) Méthode | p. 5 |
| (4) Données | p. 7 |
| (5) Formalisme SVM | p. 8 |
| (6) Formalisme RF | p. 10 |
| (7) Résultats | p. 12 |
| (8) Suite à donner | p. 14 |
| (9) SVM vs RF | p. 15 |
| (10) Machine Learning Experience | p. 16 |

La MATMUT:

- . Le Groupe MATMUT, assureur Mutualiste assure 3.2 millions de sociétaires et 6.8 millions de contrats au 31/12/16,
- . CA [2016] de 2.003 Milliards d'Euros,
- . Son cœur de métier historique : marché de l'assurance IARD, contrats automobiles et habitations.

Le contexte d'affaires:

- . Depuis 2015, sur le marché très concurrentiel de l'assurance automobile, la MATMUT, enregistre un fléchissement du nombre de contrats automobiles malgré l'établissement d'une politique tarifaire qualifiée par ses dirigeants « de commercialement agressive ».

Résultats:

- . Baisse du nombre de sociétaires automobiles,
- . Une accélération du taux de réalisation suite à l'entrée en vigueur de la Loi HAMON, depuis le 01/01/15,
- . Un ralentissement du taux d'affaires nouvelles en automobile stabilisé à 3% pour l'année 2016.

Le problème de la transformation:

- . Grâce à l'initiative DATALAB, la MATMUT souhaite travailler la problématique « business » définie comme suit :

➔ **Comment optimiser la transformation des devis commerciaux en "OUI" contrats sociétaires.**

IARD = contrat d'assurance couvrant les risques incendie, accidents et risques divers.

Depuis le 01/01/15, la loi Hamon donne aux consommateurs le pouvoir de résilier leur contrat d'assurance auto, moto et habitation à la date de leur choix, passé un an de contrat.

PROBLEME

Problème « Business »: le réseau commercial veut des réponses **automatiques** aux questions :

- quels sont les devis qui risquent de se transformer ? [étiquette : OUI]
- quels sont les devis qui risquent de ne pas se transformer ? [étiquette : NON]

L'objectif opérationnel: produire une étiquette de classification [OUI] / [NON] sur chaque devis généré tel que :

- investir du temps commercial pour transformer les prospects [OUI] ou potentiel en sociétaire,
- ne pas investir du temps commercial sur les prospects [NON].

Problème « Mathématique »:

. temps 1 : sur une base de devis sociétaires, avec des (X_i, Y_i) entraîner un classifieur à produire des Y_i _chapeau, avec paramétrisation du modèle optimal et choix optimal du classifieur selon les critères de performance de classification : matrice de confusion, Accuracy (basé sur le taux de bien classés) et AUC (ROC)

. temps 2 : sur une nouvelle base devis prospect $(X_i\text{ new})$ seulement, je peux produire des Y_i _chapeau_new, prédire l'appartenance de X_i _new à l'étiquette de classe Y_i avec $i \in [\text{OUI}, \text{NON}]$

CONCLUSION:

. Le problème de transformation de devis prospect en contrat sociétaire se formalise mathématiquement comme un problème de classification binaire supervisée qui entre dans le cadre l'apprentissage statistique supervisé (ou Machine Learning supervisé).

METHODE (1/2)

| JALON 1 | Période 1 : trim4/16 | Les livrables | Eléments de Méthode |
|---------|--|---|---|
| | <ul style="list-style-type: none"> . Appréhender le jeu de données MATMUT et prédire des premiers résultats de classification . Analyser les données (728674 x 112) . Tester 6 modèles : ACP, AFD, LDA, RL, RF, XGboost . Sélectionner le classifieur de référence | | <ul style="list-style-type: none"> . Sélectionner les X_i discriminantes . Vérifier la relation linéaire ou non entre X_i et Y_i . Identifier le modèle le plus parcimonieux $Y_i \sim X_i$ |
| JALON 2 | Période 2 : t3/17 | Les livrables | Eléments de Méthode |
| | <ul style="list-style-type: none"> . Mobiliser le classifieur alternatif SVM et vérifier si les résultats de prédiction sont améliorés ? | <ul style="list-style-type: none"> . Dataset optimisé . Produire les résultats modèle RF grid search . Produire les résultats modèle SVM grid search . Comparer résultats SVM vs RF | <ul style="list-style-type: none"> . Traiter les data : NA, centré-réduit, outlier, normalisé . Construire un échantillon optimisé (6150 x 7) - Méthode de SSP . Construire des valeurs optimisées Accuracy, AUC - RF grid search (classsifieur de référence Période 1) → construire un « point de référence » ACCURACY, AUC . Construire des valeurs optimisées Accuracy, AUC - SVM grid search . TESTs statistiques : comparer et expliquer les résultats SVM vs RF = identifier le classifieur le plus performant en prédiction pour garantir les réponses à notre problématique business. |

PLANIFICATION
PROJET

METHODE (2/2) – Grid SEARCH = PLAN D'EXPERIENCE où l'on cherche à maximiser une surface de décision

- . Méthode : avec « RF Période1 » → procédure de réglage des paramètres internes du classifieur RF (Mtry, Ntree) et SVM (C, Gamma), sur un dataset optimisé, afin d'obtenir les meilleures valeurs sur les critères de performances

Modèle 1 : RF « random search »,

+ parallèle

recherche aléatoire dans une longueur de recherche [1,5] pour le paramètre Mtry
Ntree fixé automatiquement

Modèle 2 : RF « grid search - accuracy »,

+ parallèle

+ longueur recherche [1,5]
+ autre mode de recherche consiste à définir une grille de valeurs de paramètres à tester.
pour Mtry c(1,3)
Ntree fixé automatiquement

Modèle 3 : RF « grid search - roc »,

modèle 2,

+ parallèle

+ grid search avec critère de perf. = roc

Longueur = nombre de valeurs différentes à essayer pour chaque paramètre d'algorithme.

1ère itération calcul SVM :

initialisation avec noyau radial ?
 $n = 6150 + \text{accuracy}$

. C = 1000, Gamma = $1/p = 0.17$

. Sans : Gridsearch, calcul parallèle

2e itération SVM :

4 noyaux comparés > AUC max ?
 $n = 6150 + \text{AUC}$

. C = 1000, Gamma = $1/p = 0.17$

. Sans : Gridsearch, calcul parallèle

3e itération : version (a)

4 noyaux comparés
 $n = 614 + \text{accuracy}$

. C = ?, Gamma = ?

. Sans : calcul parallèle

. Gridsearch = seq(0.3,0.6,by=0.01)

3e itération : version (b)

4 noyaux comparés
 $n = 614 + \text{accuracy}$

. C = 1000, Gamma = ?

. Sans : calcul parallèle

. Gridsearch = seq(0.15,0.20,by=0.01)

4e itération : version (a)

noyau radial
 $n = 6150 + \text{accuracy}$

. C = ?, Gamma = ?

. Calcul parallèle

. Gridsearch = seq(0.3,0.6,by=0.01)

4e itération : version (b)

noyau radial
 $n = 6150 + \text{accuracy}$

. C = 1000, Gamma = ?

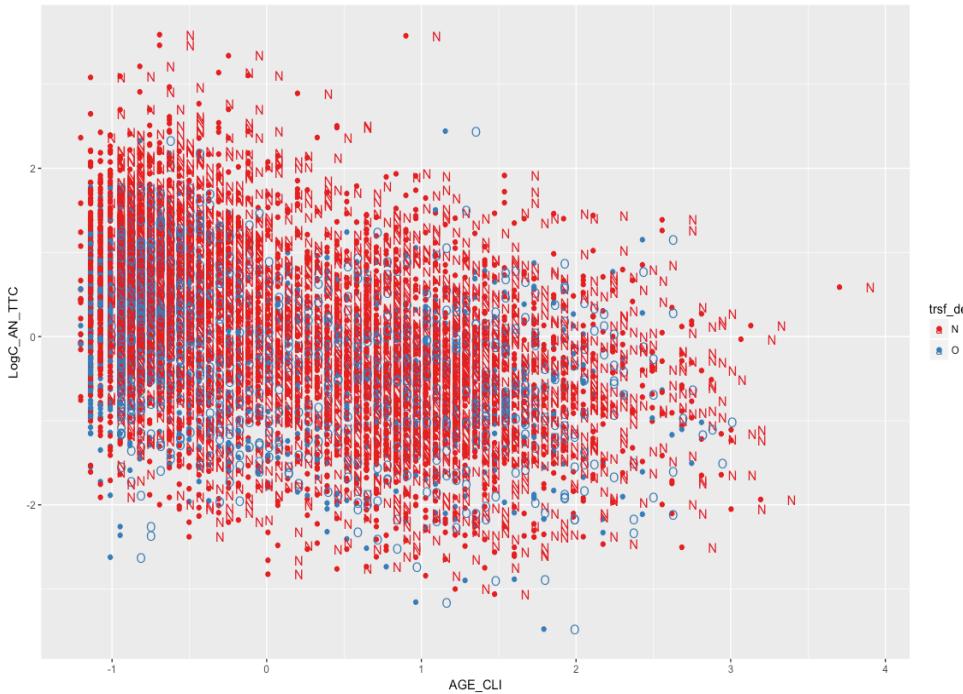
. Calcul parallèle

. Gridsearch = seq(0.15,0.20,by=0.01)

DONNEES

Classes désequilibrées: données imparfaitement séparables.

=> NON = 73.02% et OUI = 26.97%



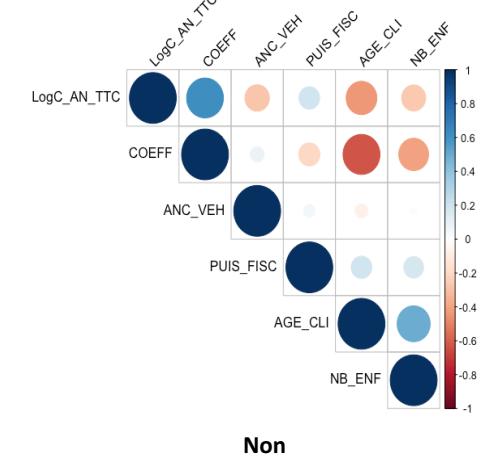
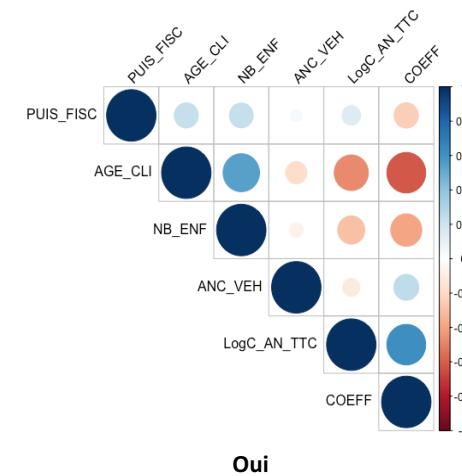
« tri croisé » exprimé en fonction des modalités de la variable d'intérêt (N, O) :

LOG Cotisation ~ Age Client (n = 6150)

Qualifier Relation Y ~ Xi: non linéaire (i.e. quadratique)

=> La matrice de variance-covariance Oui ≠ Non

(graphiquement matrice des corrélations n = 6150)



Oui

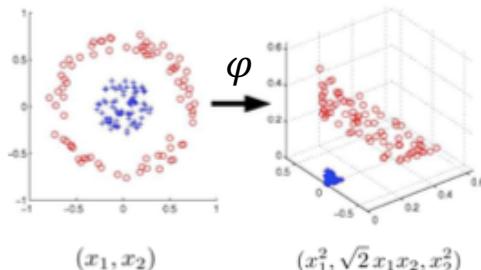
Non

. Visualisation des 2 principales spécifications du modèle des données à gérer par le classifieur optimisé.

cadre théorique

① Le noyau SVM : Comment rendre un problème linéairement séparable ? → un classifieur géométrique

■ Changement de représentation idéal



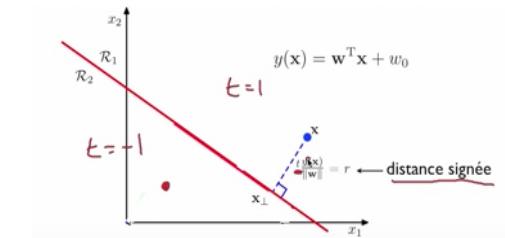
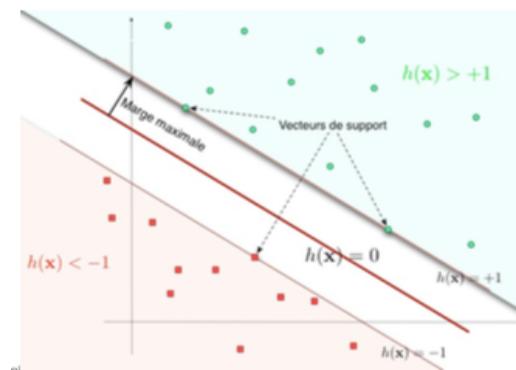
espace départ : dim = 2 axes (x,y)
problème non linéairement séparable

espace d'arrivée : dim = 3 axes (x,y,z)
Problème linéairement séparable

Appliquer une transformation non linéaire φ telle que :
 $k(x_1, x_2) = \varphi(x_1) \cdot \varphi(x_2) = f(x_1^2, \sqrt{2}x_1x_2, x_2^2)$

« la ruse du noyau = kernel trick » : augmenter la dimension des vecteurs X_i de « 1 », c'est à dire trouver une transformation non linéaire pour augmenter la dimension de l'espace initial des points-données, c'est à dire pour projeter les points-données de l'espace de départ vers un espace d'arrivée permettant de rendre les points-données linéairement séparables.

② Fonctionnement d'un SVM : Hyperplan optimal et marge maximale

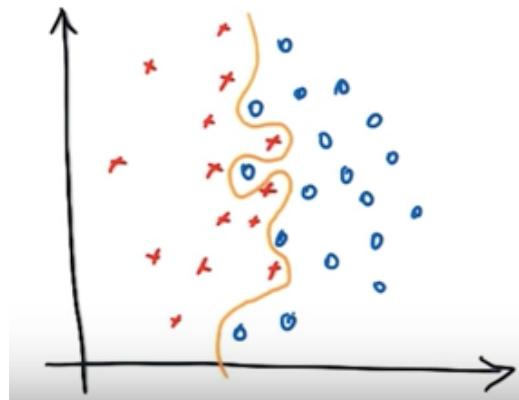


Définition : la marge est la plus petite distance signée entre la surface de décision et les entrées de l'ensemble d'apprentissage.

Définition : le meilleur Classifieur SVM est celui qui maximise la marge, c'est à dire en partant de hyperplan séparateur (droite linéaire rouge), élargir la marge (la bande aux bordures marrons) jusqu'au moment où j'arrive à la frontière de toucher les points rouges et verts.

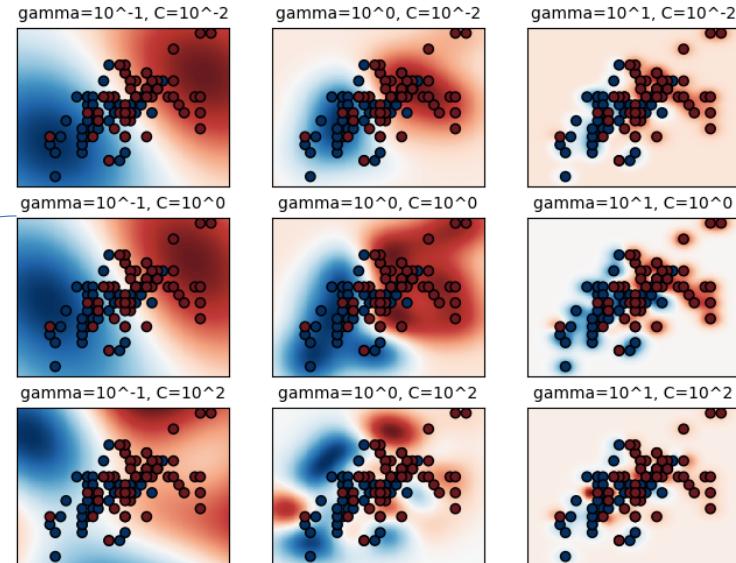
② Fonctionnement d'un SVM :

Hyperplan optimal et marge maximale



*en situation réelle - sur
un vrai jeu de données*

?



Classifieur SVM optimal :

- valeurs C, γ optimales -> celles qui permettent 1 erreur de test minimale par validation croisée,
- La marge est réduite « hard margin », et sa frontière de décision colle aux données pour mieux séparer les points.

③ Réglages :

| | PROBLEME LINEAIRE | PROBLEME NON LINEAIRE |
|-----------------------------------|--|--|
| DONNEES SEPARABLES | | <i>noyau non linéaire $K(x,x')$</i> |
| DONNEES IMPARFAITEMENT SEPARABLES | <i>Optimisation C, γ</i> | <i>Optimisation C, γ</i> |

① Définition: 2001 - Leo Breiman

- Cet algorithme appartient à la famille des agrégations de modèles (**famille des méthodes d'ensembles**)
 - c'est en fait un cas particulier de bagging (bootstrap aggregating) appliqué aux arbres binaires de décision de type CART.
- = combinaison de classifieurs simples pour obtenir un classifieur plus performant.**

Pseudo – code : (source wikipedia.fr)

```

On veut prédire  $y_0$ , la décision associée à  $x_0$ .
On a un échantillon d'apprentissage  $z = ((x_1, y_1), \dots, (x_n, y_n))$ 
Pour  $B$  allant de 1 à  $B$  faire:
    Tirer un échantillon aléatoire  $z^*$ .
    Estimer un arbre sur cet échantillon  $z^*$  avec
    randomisation des variables: la recherche de
    chaque nœud optimal est précédé d'un tirage
    aléatoire d'un sous-ensemble de  $m$  prédicteurs
Fin
Calculer l'estimation moyenne  $\varphi_B(x_0) = \frac{1}{B} \sum_{b=1}^B \varphi_{z^*}(x_0)$ ;
Si cette estimation est plus grande que  $\alpha$ , alors la décision
finale est 1; sinon, c'est 0.

```

- Permet de décorrélérer les arbres de décision générés
- Paramètres
- Bagging

Alpha α (paramètre interne du RF) : appelé « cut-off », est un prior qui permet de fixer à la probabilité 0.5 la fréquence des classes de l'étiquette lors de l'apprentissage (~ c'est un apprentissage fait avec le modèle optimal d'apprentissage, les classes sont équiprobables)

<https://www.hackerearth.com/fr/practice/machine-learning/machine-learning-algorithms/tutorial-random-forest-parameter-tuning-r/tutorial/>

② Principe:

- Le principe des méthodes de Bagging et donc en particulier des forêts aléatoires : c'est de faire la moyenne des prévisions de plusieurs modèles indépendants pour réduire la variance et donc l'erreur de prévision.

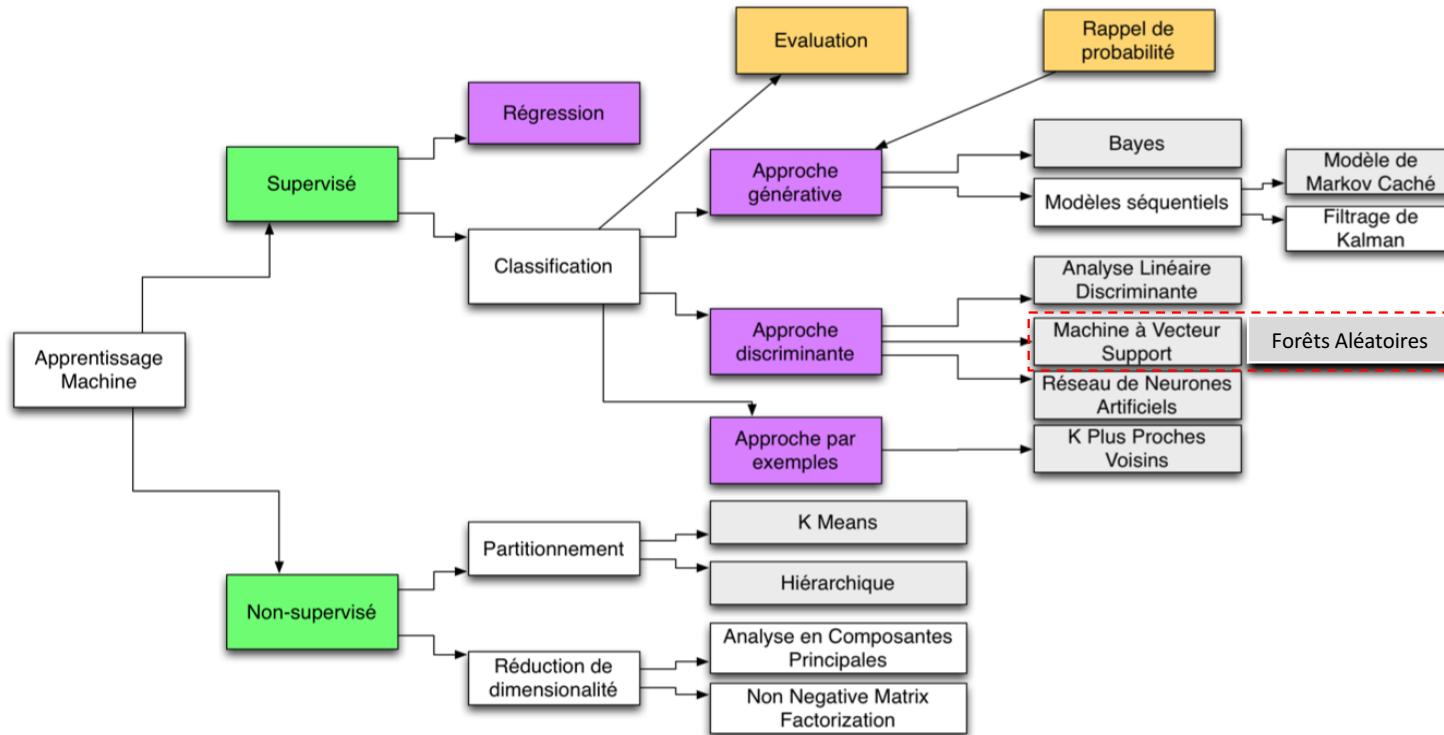
③ Réglages:

- **Aléatoire aux niveaux des individus :**
 - Pour construire ces différents modèles, on sélectionne plusieurs échantillons bootstrap de données, c'est à dire des tirages avec remises.
- **Aléatoire au niveau des variables :**
 - Pour chaque arbre on sélectionne un échantillon bootstrap d'individus et à chaque étape, la construction d'un noeud de l'arbre se fait sur un sous-ensemble de variables tirées aléatoirement.
- **Comment obtenir l'estimation finale ?**
 - On se retrouve donc avec plusieurs arbres et donc des prédictions différentes pour chaque individu.

En classification : on choisit la catégorie la plus fréquente
En régression : on fait la moyenne des valeurs prédites

Le fonctionnement interne du RF est en « boîte noire »

Taxinomie des différents modèles d'apprentissage statistiques



Source : Geoffroy Peeters - peeters@ircam.fr - 2015

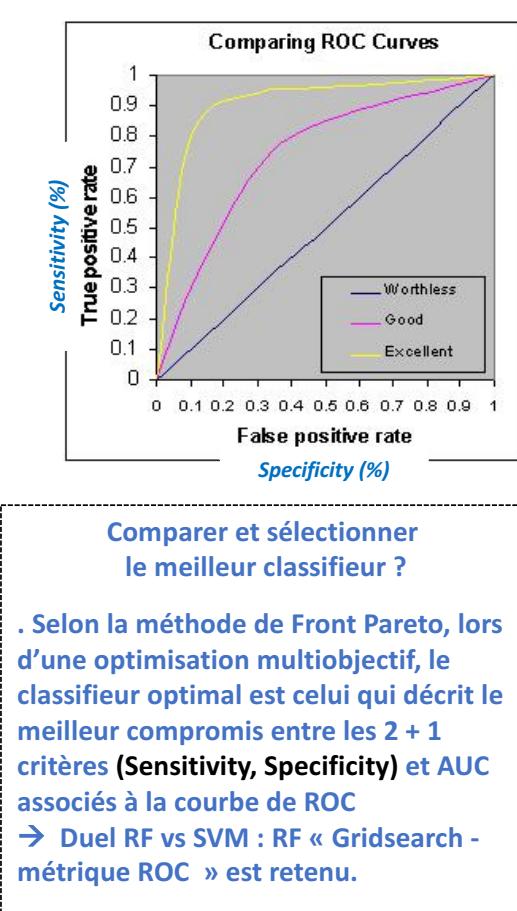
RESULTATS (1/2)

* X_i quantitatives – continues, ce résultat est aligné avec la qualité des données du Système d'Information MATMUT.

** CV = validation croisée

| Items | Résultats observés |
|------------------------------|--|
| Résultats – période1 | . 6 variables explicatives continues* sélectionnées (vs. 112 X_i du fichier de départ) |
| Le classifieur retenu | . le classifieur RF est le modèle retenu : AUC = 0.69 et F-Mesure = 0.85 |
| Variables importantes | . 2 variables discriminantes : Cotisation Annuelle et Age Client [<i>alignées avec le pricing Matmut</i>] |
| Spécification du modèle | <ul style="list-style-type: none"> . pas de relation linéaire entre les X_i et le Y . les X_i discriminantes ne sont pas toutes identifiées (tarifs concurrents) . le rôle important des classes déséquilibrées du Y |
| Résultats – période2 | . un échantillon sondage stratifié proportionnel ($n = 6150$) est suffisant pour le calcul |
| Méthodologie | . une procédure de gridsearch, CV**, calcul parallèle permet le calcul de classifieurs optimisés |
| Famille des RF | . le meilleur modèle : RF "gridsearch - ROC", AUC = 0.61 et F-mesure = 0.74 |
| Variables importantes | . Cotisation Annuelle et Age Client |
| Famille des SVM | . le meilleur modèle : SVM Gaussien $C = 1000$ et $\gamma = 0.15$, AUC = 0.56 et F-mesure = 0.74 |
| Le critère de performance | . le critère ROC-AUC (et ses critères Sensibilité et Spécificité) sélectionne le meilleur classifieur [vs] |
| Le classifieur retenu | . le Random Forest "grid search - ROC" |
| Le classifieur non retenu | . le SVM Gaussien qui ne sait apprendre que du « NON » |
| La réponse Business | . les résultats insuffisants du RF qui détecte seulement 12.27% de « OUI » |

RESULTATS (2/2) – METRIQUES PERIODE 2



| METRICS | RF "grid search - ROC" | METRICS | SVM Grid Gaussian |
|---------------------------------------|------------------------|---|-------------------|
| Tx de biens classés (Accuracy) | 0,7131 | Tx de biens classés (Accuracy) % | 0,7452 |
| Tx d'Erreur | 0,2869 | Tx d'Erreur % | 0,2548 |
| AUC | 0,6143 | AUC | 0,5685 |
| Kappa | 0,0560 | Kappa | 0,00 |
| F1 (F-mesure) | 0,7485 | F1 (F-mesure) | 0,7452 |
| Accuracy | 0,7131 | Accuracy | 0,7452 |
| 95% CI | (0,6919, 0,7337) | 95% CI | (0,7319, 0,7582) |
| No Information Rate | 0,7392 | No Information Rate | 0,7452 |
| P-Value [Acc > NIR] | 0,9946 | P-Value [Acc > NIR] | 0,5081 |
| Kappa | 0,056 | Kappa | 0 |
| McNemar's Test P-Value | <2e-16 | McNemar's Test P-Value | <2e-16 |
| Sensitivity | 0,9215 | Sensitivity | 1 |
| Specificity | 0,1227 | Specificity | 0 |
| Pos Pred Value | 0,7485 | Pos Pred Value | 0,7452 |
| Neg Pred Value | 0,3554 | Neg Pred Value | NaN |
| Prevalence | 0,7392 | Prevalence | 0,7452 |
| Detection Rate | 0,6811 | Detection Rate | 0,7452 |
| Detection Prevalence | 0,91 | Detection Prevalence | 1 |
| Balanced Accuracy | 0,5221 | Balanced Accuracy | 0,5 |
| Positive' Class | N | Positive' Class | N |

Sensitivity = sensibilité

Specificity = spécificité

Capacité à identifier les VP (N)

Capacité à identifier les VN (O)

... si on reprenez le problème du début mais différemment

Problème

- ❖ Classification supervisée
- ❖ Variable cible
 - Variable binaire [Oui; Non]
 - Classes moyennement déséquilibrées
- ❖ Relation quadratique $Y \sim X_i$
- ❖ Modèle sous-spécifié avec un défaut des X_i discriminants

Travailler les spécifications d'un modèle mieux adapté au problème

- ❖ Prédiction : viser le compromis erreur apprentissage et erreur en test
 - Axe 1 : itérer les 2 classifieurs finaux en apportant des compléments de spécifications :
 - *apporter des X_i supplémentaires : intégrer les données de tarif concurrent,*
 - *tester les modèles hybrides : classifieur avec un algorithme de redressement des classes déséquilibrées (SMOTE : Synthetic minority oversampling technique)*
 - Axe2 : itérer d'autres techniques de classification
 - *développer les modèles de détection d'événement rare (ex. Fraude & modèle QDA)*
 - *tester les réseaux de neurones sur environnement technique GPU*

RF vs SVM

Theorem : The No Free Lunch Theorem means there is no one best algorithm that does best in all cases.

Il n'y a pas de repas gratuit, il n'existe pas de super algorithme capable de traiter tous les cas d'usages sur un jeu de données.

| RANDOM FOREST | | SVM | |
|--------------------------------------|---|--------------------------------------|--|
| | AVANTAGES | | AVANTAGES |
| | INCONVENIENT | | INCONVENIENT |
| apprentissage statistique | généralise bien | apprentissage statistique | algorithme géométrique, facile à comprendre pour les problèmes de classification automatique (généralisation de l'analyse discriminante) |
| interprétabilité | classement des descripteurs | parcimonie en apprentissage | cas linéairement séparable -> peu de data pour un apprentissage semi-supervisé |
| prédiction | résultats robustes | prédiction | quand les SVM marchent, ils marchent bien |
| traitement grande base (Big Data) | ► passe la mise à l'échelle (« scalable ») | expertise accrue $Y \sim X_i$ | permet une compréhension élargie du modèles de données |
| spectre d'utilisation | utilisable sur de nombreux problèmes de classification | spectre d'utilisation | composition de noyaux pour traiter de nombreux problèmes (textes, images, vidéo,) |
| classification multi-classes | par nature | sur-apprentissage | résultats robustes et évite le sur-apprentissage (rôle de C) |
| | | transformation nature problème | non linéaire en linéaire |
| coût de l'expertise en apprentissage | ► "fonctionnement interne en boîte noire" | coût préparation | normalisation-standardisation des données |
| coût de l'expertise sur le domaine | stratégie d'élagage délicate | coût de l'expertise en apprentissage | ► difficultés de mise en œuvre (paramétrage) |
| coût computationnel | apprentissage souvent long | coût de l'expertise sur le domaine | pas d'exploitation de connaissance a priori (def. Noyau) |
| généralisation | ► sensibilité au bruit et points aberrants, aussi bien en régression / classification | coût computationnel | ► coûteux en puissance calcul (RAM) – (pb opti quadratique) |
| variable catégorielle | modèle biaisé en faveur des variables qui ont le plus de niveaux --> classement des descripteurs faussé | interprétabilité | ► pas de classement des descripteurs |
| bonnes pratiques MATMUT | classifieur efficace pour avoir un premier regard sur un problème de classification | généralisation | résultats sensibles au noyau choisi |
| | | classification multi-classes | Le traitement des problèmes multi-classes reste une question ouverte |
| | | bonnes pratiques MATMUT | classifieur SVM efficace en technique de scoring |

MACHINE LEARNING EXPERIENCE

Pour 1541.50 EUR : Installer une machine GPU pour faire de l'apprentissage profond à la maison

① configuration:



| | |
|--|-------------------------|
| CPU - Intel - Core i5-6600K Processeur Quad-Core 3,5 GHz | € 242.50 |
| RAM - G.Skill - Ripjaws Série V 32 Go (2 x 16 Go) Mémoire DDR4-2133 | € 200 |
| GPU - EVGA - GeForce GTX 1070 8 Go SC Gaming Carte vidéo ACX 3.0 | € 589 |
| SSD - Samsung - 850 EVO-Series 500Go 2.5" Solid State Drive | € 150 |
| Carte mère - MSI - Z270-A PRO ATX LGA1151 mère | € 140 |
| CPU Cooler - Cooler master - Hyper 212 EVO 82,9 CFM manches CPU roulement Cooler | € 30 |
| alimentation - EVGA - SuperNOVA NEX 650W 80+ Gold Certified Fully Modular ATX Power Supply | € 80 |
| Boîtier - Corsair - Étui Tower Tower ATX 200R | € 110 |
| | Total: € 1541.50 |

② logiciels:

Plateforme
Data Fabric ?

| | |
|---|--------|
| OS - Ubuntu 16.04 | € 0.00 |
| Drivers GPU | |
| CUDA (v8) - une plate-forme informatique parallèle en tirant parti de la puissance du GPU | € 0.00 |
| CUDNN (v6) - Bibliothèque CUDA Deep Neural Network qui se trouve sur le dessus de CUDA. | € 0.00 |
| BIBLIOTHEQUES DE CALCUL | |
| Tensorflow (v1.3) - cadre d'apprentissage machine Google | € 0.00 |
| Theano (v0.9) - Un cadre alternatif d'apprentissage de machine. | € 0.00 |
| Keras - bibliothèque de réseau neuronal de niveau supérieur qui s'exécute sur Tensorflow, Theano. | € 0.00 |
| IDE | |
| Anaconda (v3.6) + Python (3.5) + R (3.41) | € 0.00 |
| Total: € 1541.50 | |

⚠ Bonnes Pratiques DATALAB:

INSA INSTITUT NATIONAL
DES SCIENCES
APPLIQUÉES
ROUEN

→ [NVIDIA GPU Research Center](#)

 **NVIDIA** ACCELERATED COMPUTING

- ANNEXE TECHNIQUE -

① Objectif: construire un dataset optimisé

- Obtenir des calculs R qui tiennent en RAM ordinateur et un temps de traitement réduit pour les SVM,
- Apprentissage statistique : attention à l'optimisation du compromis, temps de calcul et précision des résultats en particulier en échantillonnant les données pour réduire les temps de calcul afin de mieux optimiser les modèles.

② Principe:

- Construire un sondage stratifié proportionnel sur la variable d'intérêt Y_i (`trsf_dev`),
- Dans notre cas on prélève 1% de l'effectif total 614929 lignes, c'est à dire un effectif $n = 6150$ lignes,
- La taille prélevée de $n = 6150$ lignes n'est ni bonne, ni mauvaise. Simplement, par empirisme, il a été testé que les calculs du code SVM / RF sur cette taille tiennent en « Ram » sans générer des temps de calculs trop longs.

➔ **Preuve:** Lire les temps de calcul enregistrés dans les grilles de recherche de l'optimisation des paramètres du SVM.

③ Formalisme de la théorie des sondages:

- **Méthode sondage stratifié optimal ou sondage stratifié proportionnel ?**
- L'outil de sondage stratifié en utilisant la variable intérêt Y_i [O/N] (`trsf_dev`) comme variable de stratification est adapté à notre dataset et problème associé (i.e. conforme aux recommandations de JM Poggi sur l'utilisation de classifieur avec échantillonnage),
- La méthode de sondage stratifié optimal est non nécessaire ici, car les variables explicatives X_i sont centrées-réduites ($\mu = 0$, $\sigma = 1$),
- Il faut s'assurer du prérequis suivant : vérifier que les « metrics » moyenne et écart-type sont bien respectées au sein des sous populations Non / Oui de l'échantillon prélevé.

➔ **Preuve:** pas d'introduction de biais de sélection suite à un prélèvement de 6150 lignes.

RAPPELS :

- Les données BRUTES sont bruitées et déséquilibrées,
- Classification bi-classes supervisées : déséquilibre minoritaire quand une classe est minoritaire de [10;20] % de la classe la plus élevée,
- Les classifieurs sont utilisés pour trier une masse énorme de cas négatifs « VN » et trouver un petit nombre de cas positif « VP »,
- Les classifieurs sont biaisés vers la classe majoritaire car leurs fonctions de perte optimisent le taux d'erreur, sans tenir compte de la distribution des données,
- L'algorithme d'apprentissage génère simplement un classificateur trivial qui classe chaque exemple comme classe majoritaire.

QUESTIONS :

- Que dois-je faire quand mes données sont déséquilibrées?
- **Quel algorithme d'apprentissage est le meilleur?**

AXES DE REEQUILIBRAGE :

- Ne fais rien. Parfois, vous avez de la chance et rien ne doit être fait.
- Équilibrer l'ensemble d'entraînement d'une manière ou d'une autre:
 - Sur-échantillonner la classe minoritaire,
 - Sous-échantillonner la classe majoritaire,
 - **Synthétiser de nouvelles classes minoritaires.**
- Jetez les exemples minoritaires et passez à un cadre de détection des anomalies.
- Au niveau de l'algorithme, ou après:
 - **Ajuster le poids de la classe** (coûts de mauvaise classification ~ classe minoritaire),
 - Ajuster le seuil de décision,
 - Modifier un algorithme existant pour être plus sensible aux classes rares,
- Construire un algorithme entièrement nouveau pour bien fonctionner sur des données déséquilibrées.

LES METRIQUES A UTILISER :

- Le ROC-AUC,
- Le score F1 (F-mesure),
- Kappa de cohen (« accuracy » en déséquilibré).

LES IMPLEMENTATIONS : « non retenues »

- L'argument bayésien : (wallace et al.)
→ sous-échantillonner la classe majoritaire
- approches basées sur les voisins
→ Suppression des liens de Tomek (recherche de proximité dans la paire instance majoritaire-minoritaire puis suppression de la majoritaire dans la paire)

SMOTE : Synthetic Minority Oversampling TEchnique

- Le principe : échantillonner la classe majoritaire et synthétiser de nouvelles instances minoritaires en interpolant celles qui existent déjà, (sur-échantillonner la classe minoritaire en créant des exemples synthétiques plutôt qu'en sur-échantillonnant avec remplacement),

Pseudo-code :

- Étape 1 : la première étape ignore les exemples de la classe majoritaire,
- Étape 2 : pour chaque instance minoritaire, choisissez ses K plus proches voisins,
- Étape 3 : créer de nouvelles instances à mi-chemin entre la première instance et ses voisins,

MOBILISER PLUS DE DESCRIPTEURS:

- Acheter ou créer plus de Xi,
- **Transformer le problème avec des variables proxy et latentes,**
- Utiliser l'apprentissage par transfert pour apprendre un problème et transférer les résultats à un autre problème.

Résultats : Rééquilibrage SMOTE + SVM de référence

| Etiquette Yi = trsf_dev | NON | OUI |
|-------------------------------|-------|-------|
| répartition dataset (n= 6150) | 73,31 | 26,69 |
| avec rééquilibrage SMOTE | 50,00 | 50,00 |

Famille des RF . le meilleur modèle : RF "gridsearch - ROC", AUC = 0.61 et F-mesure = 0.74

SMOTE + SVM référence . le meilleur modèle : SVM Gaussien C = 1000 et $\gamma = 0.18$, AUC = 0.5986 et F-mesure = 0.7384

SVM référence (old) . le meilleur modèle : SVM Gaussien C = 1000 et $\gamma = 0.15$, AUC = 0.56 et F-mesure = 0.74

```

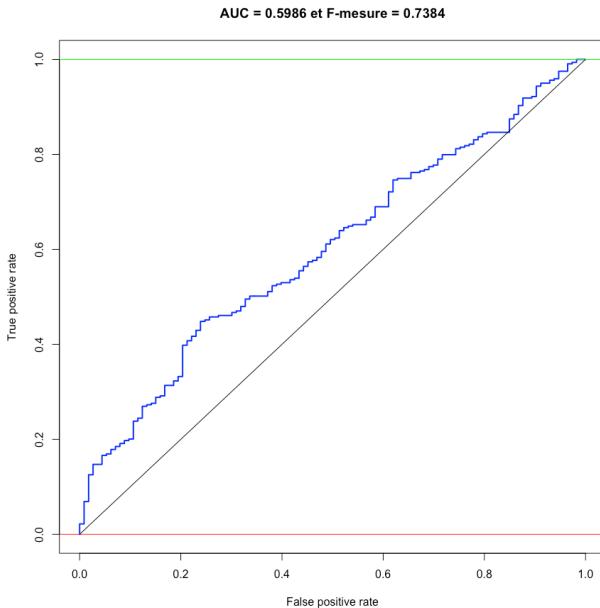
Accuracy : 0.7384
95% CI : (0.6943, 0.7793)
No Information Rate : 0.7384
P-Value [Acc > NIR] : 0.5253

Kappa : 0
McNemar's Test P-Value : <2e-16

Sensitivity : 1.0000
Specificity : 0.0000
Pos Pred Value : 0.7384
Neg Pred Value : NaN
Prevalence : 0.7384
Detection Rate : 0.7384
Detection Prevalence : 1.0000
Balanced Accuracy : 0.5000

'Positive' Class : N

```



ANALYSE :

- . Le rééquilibrage de classe avec l'algorithme SMOTE fonctionne,
- . Comme les classes sont rééquilibrées à 50.00%, la valeur optimale du γ (gamma) est plus forte (0.18 au lieu de 0.15 – alignée avec la théorie),
- . En terme AUC et F-mesure, les valeurs du SVM SMOTE sont proches de celles du Modèle RF « gridsearch - ROC »,

Conclusion :

- . le rééquilibrage de classes à l'aide de l'algorithme SMOTE pour le classifieur SVM joue le même rôle « que le prior cut-off » du modèle Random Forest.
- . Au final, pour Random Forest et SVM, on obtient le classifieur optimisé dans sa famille de modèle.