Devoir surveillé de classification supervisée : apGIS4

Vincent Vandewalle

04/04/2019

Durée 2h, tous documents autorisés

Présentation des données

On souhaite prédire la souscription à une assurance lors d'un vol aérien. Les variables sont les suivantes : - Target: Claim Status (Claim) - Name of agency (Agency) - Type of travel insurance agencies (Agency.Type) - Distribution channel of travel insurance agencies (Distribution.Channel) - Name of the travel insurance products (Product.Name) - Duration of travel (Duration) - Destination of travel (Destination) - Amount of sales of travel insurance policies (Net.Sales) - Commission received for travel insurance agency (Commission) - Gender of insured (Gender) - Age of insured (Age)

La variable à prédire est la variable `Claim`

Importation des données et premières analyses

Q1 : Importer les données à partir du fichier `assurance.csv`. On nommera `don` le data.frame résultant. Le jeu de données comporte-t'il des valeurs manquantes ? Quel option doit-on préciser dans R pour préciser la chaîne de caractères associée aux valeurs manquantes ?

R1 :

Par la suite on chargera le fichier `don.Rda` contenant le data.frame `don` pour être sûr de partir sur de bonnes bases.

Q2 : En quoi le problème qui vous est posé est-il un problème de classificiation supervisée ? Quel intérêt peut-il bien y avoir à prédire la variable `Claim` ?

R2 :

Q3 : Dans vos données quelles sont les fréquences des différentes modalités de la variable `Claim` ? Dans quel ordre des modalités de la variable `Claim` sont-elles codées ?

R3 :

Q4 : Pouvez-vous donner une règle de classement qui a un taux de bon classement supérieur à 98%

R4 :

Q5 : En utilisant jusdicieusement les fonctions `sapply` et `nlevels` donner le nombre de modalités de chacune des variables. Que dire de la variable `Destination` ?

R5 :

Q6 : Réaliser un test statistique permettant répondre à la question d'existance d'un lien entre la variable `Claim` et la variable `Agency`. Que conclure ?

R6 :

Q7 : Dans la continuité de la question précédente afficher le vecteur contenant `P(Claim = Yes |Agency = x)` pour chacune des valeurs `x` de la variable agency, et la stocker dans une variable nommée `vecteur_proba`.

R7 :

Q8 : Les probabilités précédentes pourraient faire office de score, en associant à chaque individu la probabilité `P(Claim = Yes |Agency = x)` où `x` est la modalité dont dispose l'individu pour la variable `Agency`. En utilisant de manière adéquate l'indexation par nom, et en lançant une commande du type `vecteur_proba[don$Agency]`, obtenir le vecteur contenant les probabilités pour chaque individu. Enfin traçer la courbe ROC associée à ce score.

R8 :

Q9 : A votre avis si on avait effectué une régression logistique en lançant : `glm(Claim ~ Agence, family = "binomial", data = don)`, aurait-on obtenu les mêmes résultats ? Justifier.

R9 :

Q10 : On décide maintenant d'ajuster une régression logistique en prenant en compte toutes les variables. Après avoir lancé la commande adaptée on obtient les messages : a. `glm.fit: l'algorithme n'a pas convergé`, b. `glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1` De plus en analysant les sorties c. `45107 observations deleted due to missingness`

Pour le message b. il s'agit du fait que dans la formule du calcul `P(Y = 1|X = x)` l'ordinateur indique que pour certains individus cette probabilité est ajusté numériquement à 0 ou 1. Ce n'est pas un problème en soi, mais cela peut parfois être un signe d'overfitting.

Pour le message a. pourquoi parle-t'on d'algorithme ici ?

Pour le message c. expliquer précisement pourquoi 45107 observation on été supprimée. Cela est-il bien grâve ?

R10 :

Q11 : On décide de récoder la variable `Gender` comme suit :

```
don$Gender = as.character(don$Gender)
don$Gender[is.na(don$Gender)] = "UNKNOWN"
don$Gender = factor(don$Gender)
table(don$Gender)
```

Que fait le code précédent, et quel intêret pour la suite peut-il bien avoir à effectuer ce recodage ?

R11 :

Q12 : On relance maintenant l'ajustement du modèle complet (le code peut maintenant mettre un peu de temps à tourner au plus 2 min, sauvegarder avant de lancer ...). Tracer la courbe ROC associée et donner l'AUC.

R12 :

Q13 : Ici les classes sont en effectif très déséquilibrés. Les stratégies usuelles peuvent alors être : a. sous-échantillonner la classe en plus grand effectif b. sur-échantillonner la classe en plus faible effectif c. prendre directement en compte des poids dans l'ajustement du modèle, par exemple en prenant des poids inversement proportionnels à l'effectif des différentes classes

Les statégies a. et b. peuvent être mise en oeuvre en amont de l'ajustement du modèle, tandis que la stratégie c. nécessite l'utilisation de modèles capables de prendre en compte des poids.

D'un point de vue informatique la stratégie a. est la moins coûteuse en temps de calcul, mais elle conduit à perdre des données ... La stratégies b. conduirait à démultiplier le nombre de données ... Enfin la stratégie c. présenterait un coût informatique identique à l'ajustement du modèle classique (modula la prise en compte de poids)

En fait, quel intérêt peut-il bien avoir dans toutes ces stratégies par rapport à pas tenir compte du tout de ce déséquilibre ?

R13 :

Q14 : Ici, compte-tenu de la faible capacité informatique dont nous disposont pour aujourd'hui on opte pour le sous-échantillonnage.

Fixer la graine du générateur de nombres pseudo-aléatoires à la valeur 1234.

Créer un nouveau jeu de données `don_us` comme "don under-sampled", constitué des 927 individus possédant la modalité `Yes` pour la variable `Claim` et de 927 individus tirés au hasard et sans remise parmi les 62399 individus possédant la modalité `No`. On pourra bien sûr, faire des selections pour distinguer les lignes avec `Yes` de celle avec `No`, utiliser la fonction sample, ainsi que la fonction`rbind` !

R14 :

Par la suite en cas de problème l'objet `don_us` est contenu dans le fichier `don_us.Rda` au besoin.

Q15 : Ajuster maintenant le modèle de régression logistique à partir de toutes les variables sur le data.frame `don_us`. En regardant maintenant la sortie on obtient le message : `Coefficients: (9 not defined because of singularities)`, cela indique en particulier que dans la matrice du modèle, certaines colonnes peuvent être déduites comme combinaison linéaire d'autre colonnes.

Si on stocke le résultats l'ajustement précédent dans le variable `reg_us`, on peut récupérer la matrice du modèle comme suit :

```
X = model.matrix(reg_us)
```

On s'intéresse aux valeurs propres de XX^T (matrice à inverser en régression linéaire, et à peu de chose prés celle qui nécessite d'être inversée en régression logistique ...)

```
eigen(t(X) %*% X)$values
```

Que nous indique des valeurs propres trés faibles quand à notre objectif d'inversion de cette matrice ?

R15 :

Q16 : A l'aide de la commande suivante on réalise une sélection pas à pas :

```
min.model <- glm(Claim ~ 1, family = "binomial", data = don_us)
max.model <- glm(Claim ~ ., family = "binomial", data = don_us)
best.model = step(min.model, direction='both',
                  scope= list(lower = min.model,
                              upper = max.model))
```

A quoi sert l'option direction = 'both' ? Quel intérêt dans la recherche peut-il y avoir à partir du modèle le plus simple ?

R16 :

Q17 : Interpréter le modèle retenu et évaluer ses performances (Courbe ROC, seuils effectuant le meilleur compromis sensibilité / spécificité, et sensibilité et spécificité associées)

R17 :

Q18 : On souhaite ajuster un modèle de forêts aléatoires à l'aide de la commande :

```
library(randomForest)
randomForest(Claim ~ ., data = don_us)
```

On obtient le message d'erreur `Error in randomForest.default(m, y, ...) : Can not handle categorical predictors with more than 53 categories.`

Adapter le code pour pouvoir ajuster les random forests, et evaluer les performances de celles-ci.

R18 :

Q19 : On aurait pu vouloir ajuster un modèle d'analyse discriminante probabiliste. Pourquoi ne peut-on pas utiliser ici la LDA ou la QDA ?

R19 :

Q20 : Ici on souhaite ajuster un classifieur de Bayes naïf à l'aide de la fonction `naiveBayes` du package `e1071`. Réaliser cet ajustement sur le jeu de données `don`, que remarquez vous sur le temps d'excusion par rapport au temps précédent ? Comment expliquez-vous celà ?

R20 :

*Q21 :** Le classifieur de Bayes naïf (cas particulier de méthode d'analyse discriminante probabiliste) est basé sur l'estimation des $P(Y = i) = \pi_i$ et $P(X = x|Y = i) = f_i(x)$, puis de l'application du théorème de Bayes.

En passant de `don` à `don_us`, comment l'estimation de π_0 et π_1 est-elle modifiée ? Les $f_0(x)$ et $f_1(x)$ estimés sont-ils foncièrement différents entre `don` et `don_us` ? Justifier.

R21 :

Q22 : En partant de la formule de Bayes, montrer que la probabilité $P(Y = 1|X = x)$ est une fonction croissante du rapport $f_1(x)/f_0(x)$.

R22 :

Q23 : En déduire que les valeurs estimées de π_0 et π_1 n'influent en rien sur l'ordre dans lequel seront rangés les individus en terme de probabilités.

R23 :

Q24 : Conclure des question Q21 à Q23 que la courbe ROC est théoriquement inchangée entre les version répondérées ou non pour le cas de l'analyse discriminate probabiliste, dans le cas où les échantillons sont de grandes tailles (considérations asymptotiques oblige . . .)

R24 :

Q25 : En conséquence expliquer pourquoi on a tout intérêt à plutôt utiliser `don` que `don_us`

R25 :

Q26 : Parmis les évaluations réalisées pour les différents modèles, les classer de la plus sujette à une sur-évaluation des performances (biais d'optimisme), à celle la mois sujette à cette sur-évaluation. Préciser éventuellement si certaines ne sont pas sujettes du tout à ce biais.

R26 :

Q27 : Proposez une approche permettant de comparer "en toute honnêteté" les différents modèles proposés (implémentation non demandée)

R27 :