

TP3 HPC-BigData 2022 : Régression logistique

Prévision statistique d'un dépassement de seuil d'Ozone

Description des données :

L'objectif de ce TP, exploitant le fichier de données **DataTP.txt** du TP2, est d'estimer un modèle de discrimination (**régression logistique**) pour prévoir à partir des prévisions non post-traitées du modèle MOCAGE, non plus la concentration d'ozone **O3o** mais **le dépassement du seuil d'ozone de 180 µg/m³**.

On dispose dans le fichier **DataTP.txt** de mesures de concentration d'ozone **O3o** réalisées lors des étés 2002 à 2005 et des prévisions associées issues du modèle MOCAGE, classées par ordre chronologique, des 7 prédicteurs potentiels suivants :

O3p : [O₃] prévue par MOCAGE à l'échéance considérée (µg/m³)
TEMPE : Température prévue par MOCAGE pour l'échéance considérée (°C)
RMH2O : Rapport de mélange prévu par MOCAGE pour l'échéance considérée (g/kg)
FF : Force du vent prévue par MOCAGE pour l'échéance considérée (m/s)
NO2 : [NO₂] prévue par MOCAGE à l'échéance considérée (µg/m³)
JJ : Jour de la semaine (facteur à deux modalités codées S pour jours ouvrés, F pour fins de semaine et jours fériés)
STATION : Nom de la station, facteur à cinq modalités codées Aix, Cad, Pla, Ram et Als.

1. Phase préliminaire :

- Charger les données et les packages :
`data=read.table("DataTP.txt", header=TRUE)`
`library(MASS) ; library(verification)`
- Ajouter à la data.frame **data** deux nouvelles variables, **OCC** et **OCCp**, de type **factor**, pour respectivement l'occurrence observée de dépassement du seuil et l'occurrence prévue par MOCAGE. Les occurrences seront codées 1 et les non-occurrences 0.
- Exécuter le script **scores.R** qui permet d'exploiter la fonction **scores** calculant certains scores définis à partir d'une table de contingence (voir annexe).

2. Régression logistique : modèle de discrimination

- Estimer le modèle de régression logistique exploitant tous les prédicteurs potentiels (sans exploiter *O3o*) : `glm.out=glm(OCC~.,data[, -2],family=binomial)`
- Faire une sélection automatique des prédicteurs en exploitant l'indice *BIC* : `glm.outBIC=stepAIC(glm.out,k=log(nrow(data)))`
- Evaluer avec la fonction *scores* les prévisions non post-traitées d'occurrences du dépassement de seuil (variable *OCCp*) puis celles issues de la régression logistique *BIC*. Les probabilités d'occurrences prévues sur apprentissage sont disponibles ainsi : `predict(glm.outBIC,type="response")` ou `fitted(glm.outBIC)`
Comment les exploiter pour en déduire la prévision d'occurrence ? (voir annexe)
- En exploitant la courbe **ROC** (fonction *roc.plot*), déterminer le seuil de probabilité à utiliser dans le but de maximiser le score PSS.
Comparer avec les scores obtenus par exploitation du seuil 0.5
- Estimer un modèle linéaire gaussien pour prévoir directement la concentration *O3o* (sans exploiter *OCC*) puis en déduire une prévision d'occurrence de dépassement du seuil. Cette autre stratégie mène-t-elle à de bonnes performances ? Conclure.

Annexe : complément sur les scores

SCORES ELABORES A PARTIR D'UNE TABLE DE CONTINGENCE :

On note A l'occurrence du phénomène A, et NA sa non-occurrence.

PREVU	OBSERVE	
	A	NA
A	a	b
NA	c	d

Taux de succès global :
 $(a+d)/(a+b+c+d)$. Pas forcément
un bon indicateur (si A rare).

H taux de bonnes prévisions, F taux de fausse alerte :

$$H = \frac{a}{a+c}$$

$$F = \frac{b}{b+d}$$

Score de Peirce $PSS = H - F$ $-1 \leq PSS \leq 1$.

COURBE ROC : Scores F en abscisses et H en ordonnées

