TP 1

Logiciel R et prédiction de l'efflorescence algale

8 janvier 2017

1 Régression linéaire multiple

1.1 Calcul des coefficients de la régression

Question 1 : Si une variable catégorielle contient K niveaux, la commande *lm* de R la remplace par $K-1$ variables binaires correspondant à chacun des $K-1$ premiers niveaux. Ces variables créées prennent la valeur 1 si l'observation possède le niveau étudié et 0 sinon, et sont donc bien directement exploitables en tant que variables explicatives dans une régression.

Un seul des niveaux n'est pas transformé en une variable binaire afin d'éviter une colinéarité parfaite entre ces nouvelles variables. En effet, dans le cas contraire, la somme de ces variables binaires serait constante égale à 1. La matrice X'X ne serait alors pas inversible, et on se pourrait ainsi pas calculer l'estimateur des Moindres Carrés Ordinaires :

$$\hat{\beta}^{MCO} = (X'X)^{-1}X'Y$$

Question 2 : La qualité d'ajustement par un modèle linéaire peut être mesurée de différentes façons. En général, elle est mesurée par le R^2 (ou *Multiple R-squared* dans R) ou le R^2 ajusté (ou *Adjusted R-squared* dans R). L'avantage du R^2 ajusté est de ne pas être croissant en fonction du nombre de variables du modèle. Cela permet de comparer des modèles dont le nombre de variables explicatives est différent et en fait un critère de sélection de modèles.

$$R^2 = \frac{SCE}{SCT} = 1 - \frac{SCR}{SCT}$$

Où :

- SCT : Somme des carrés totale, i.e $||Y - \overline{Y}||^2$
- SCE : Somme des carrés expliquée, i.e $||\hat{Y} - \overline{Y}||^2$
- SCR : Somme des carrés des résidus, i.e $||Y - \hat{Y}||^2$

On a effet la relation : $SCT = SCE + SCR$
Le R^2 indique ainsi la part de la variance de Y expliquée par le modèle.

$$R^2_{adj} = 1 - \frac{SCR/(n-p-1)}{SCT/(n-1)}$$

Le R^2_{adj} incorpore ainsi une pénalisation par rapport au nombre de variables p du modèle étudié.

1.2 Test ANOVA

Question : D'après la table ANOVA obtenue à l'aide de la commande *anova* de R, trois variables semblent inutiles pour prévoir la variable *a1* : *season*, *Chla* et *NH4*. En effet, l'hypothèse H_0 de nullité du coefficient (ou, s'il s'agit d'une variable catégorielle, de nullité simultanée de tous les coefficients situés devant chaque variable binaire associée) n'est pas rejetée au niveau de risque 10 % dans chacun de ces cas.

Néanmoins, ce résultat est à nuancer car il dépend fortement de l'ordre dans lequel les tests sont effectués, d'autant plus que cet ordre est ici choisi arbitrairement par R. En effet, la commande *anova* de R réalise des tests de Fisher à la suite pour chaque variable étudiée en comparant ce modèle par rapport au modèle restreint contenant l'ensemble des variables situées plus haut dans la table (et non par rapport au modèle complet ou au modèle contenant uniquement une constante). Les résultats issus de cette façon de procéder

les tests ANOVA sont donc difficilement interprétables, et ne peuvent pas constituer un critère de sélection des variables explicatives du modèle.

Notons enfin l'avantage d'utiliser des tests ANOVA par rapport aux tests de Student de nullité des coefficients : cela permet d'évaluer l'impact des variables catégorielles en tant qu'une seule et même variable, et non en tant que plusieurs variables binaires.

1.3 Stepwise regression

L'AIC, ou critère d'information d'Akaike, est une mesure possible de la qualité d'un modèle statistique qui permet de le pénaliser en fonction de son nombre de paramètres p.

$$AIC = 2p - ln(L)$$

Où : L est le maximum de la fonction de vraisemblance du modèle.

L'AIC étant décroissant en fonction de la log-vraisemblance et croissant en fonction du nombre de paramètre, on cherche donc à le minimiser.

On effectue ici une régression Backward afin de sélectionner les variables explicatives à conserver. On part ainsi du modèle complet contenant l'ensemble des variables explicatives et on supprime une à une les variables du modèle afin de minimiser l'AIC. A chaque étape, on supprime la variable pour laquelle l'AIC du nouveau modèle (le modèle sans cette variable) est le plus faible. On réitère ce processus jusqu'à ce que supprimer une variable supplémentaire fasse réaugmenter l'AIC.

Question 1 : Les variables retenues par le modèle *step* sont : *size*, *mxPH*, *mnO2*, *NH4* (variable pourtant non retenue en utilisant la table ANOVA) et *PO4*.

Question 2 : La qualité d'ajustement a augmenté par rapport au modèle initial. En effet, le R^2 ajusté est maintenant de 0.3325 contre 0.3204 pour le modèle contenant toutes les variables. Cependant, la qualité d'ajustement n'a augmenté que de manière modeste (à peine plus de 0.01). Il semble peu probable que ce modèle ait une capacité prédictive véritablement meilleure que le modèle initial. Sans suprise, on remarque également que le R^2 "classique" a diminué par rapport au modèle complet, puisqu'on a supprimé des variables.

1.4 Conclusion sur la sélection de variables

Le TP permet ainsi d'étudier 3 façons de sélectionner les variables explicatives à introduire dans le modèle de régression linéaire multiple : le R^2 ou le R^2_{adj}, les test ANOVA de nullité des coefficients, ou bien la régression Stepwise. L'avantage de cette dernière méthode par rapport aux deux précédentes est qu'elle n'est pas spécifique aux modèles de régression linéaire multiple. L'AIC est en effet un critère qui peut être utilisé pour sélectionner les variables dans d'autres types de régressions, du moment que l'on peut estimer le maximum de vraisemblance.

2 Arbre de décision

2.1 Prévisions des observations de test

Table 1 – Résultats pour la variable a1 sur la base test

a1	MAE	MSE	RMSE	MAPE	NMSE	NMAE
Régression linéaire	12.52	276.44	16.63	Inf	0.6582	0.7822
Arbre de décision	11.65	285.74	16.90	Inf	0.6803	0.7279

```
summary(test.algae)
test.algae = knnImputation(test.algae, k =10, meth = "median")

lm.predictions.a1 = predict(final.lm, test.algae)
rt.predictions.a1 = predict(rt.a1, test.algae)

regr.eval(algae.sols[, "a1"], rt.predictions.a1, train.y = algae[,"a1"])
regr.eval(algae.sols[, "a1"], lm.predictions.a1, train.y = algae[,"a1"])
```

Commentaire sur les résultats : Les résultats obtenus sur l'échantillon de test sont globalement mauvais. Les mesures d'erreurs (*MAE* et *RMSE*) sont similaires sur la base d'apprentissage et sur la base test pour le modèle de régression linéaire mais bien plus élevées sur la base test que sur la base d'apprentissage pour l'arbre de décision. Il semble donc y avoir un phénomène de surapprentissage lorsqu'on utilise l'arbre de décision. Alors que les mesures d'erreurs obtenues à partir d l'arbre de décision étaient plus faibles que celles du modèle linéaire sur l'échantillon d'apprentissage, elles sont désormais similaires sur l'échantillon de test. La faible qualité des prédictions des deux modèles est confirmée par les graphiques.

Commentaire sur les mesures d'erreur analysées : On note ici que l'erreur MAPE n'est pas interprétable ici, car certaines vraies valeurs de $a1$ sont exactement nulles. Ceci explique d'ailleurs que R renvoie la valeur $+\infty$. Pour rappel, l'erreur MAPE ou Mean Absolute Percentage Error est donnée par :

$$\frac{100}{n}\sum_{i=1}^{n}\left|\frac{Y_i - \hat{Y}_i}{Y_i}\right|$$

Notons également les mesures d'erreur NMAE (Normalized Mean Absolute Error) et NMSE (Normalized Mean Squared Error) qui expliquent l'utilisation du paramètre train.y dans le code R, indiquant le vecteur des valeurs de la variable cible dans la base d'apprentissage utilisée pour entraîner le modèle analysé. Les mesures NMAE et NMSE comparent en effet les erreurs du modèle analysé par rapport à un modèle de référence : le modèle avec une constante pour seule variable explicative. Ainsi, si ces mesures d'erreur sont supérieures à 1, cela signifie que les prédictions obtenues sont pires que celles qui seraient obtenues avec ce modèle de référence. On remarque ainsi que sur la base d'apprentissage, $NMSE = 1 - R^2$.

2.2 Résultats pour les 6 autres variables

On remarque tout d'abord que les variances et les moyennes sur la base d'apprentissage des fréquences de ces différentes algues sont assez différentes (Tableau 2). Il est ainsi impossible de comparer directement la qualité de prédiction de 2 modèles pour 2 variables différentes, notamment en utilisant des mesures d'erreurs telles que le MAE, MSE, RMSE et RMAE qui ne sont pas normalisées.

Table 2 – Moyennes et écart-types pour les différents types d'algues

	Algue 1	Algue 2	Algue 3	Algue 4	Algue 5	Algue 6	Algue 7
Moyenne	16.92	7.46	4.31	1.99	5.06	5.96	2.50
Ecart-type	21.35	11.03	6.95	4.42	7.50	11.66	5.16

Comparaison des modèles de régression : Il est important de noter que les variables retenues dans la régression stepwise sont différentes pour chaque algue. Le tableau 3 présente ces variables (les variables sélectionnées dans la régression stepwise sont indiquées par une étoile) ainsi que les R^2 ajustés pour chacune des régressions.

Table 3 – Variables retenues dans les différentes régressions *stepwise*

	Algue 1	Algue 2	Algue 3	Algue 4	Algue 5	Algue 6	Algue 7
Season			*		*	*	
Size	*		*		*	*	*
Speed		*	*		*	*	
mxPH	*	*		*			*
mnO2	*		*	*	*	*	*
Cl						*	*
NO3				*	*	*	*
NH4	*		*	*	*	*	*
PO4	*			*	*	*	
oPO4						*	
Chla		*					
R^2 ajusté	0.3325	0.1968	0.1253	0.3034	0.2276	0.3533	0.066

Les modèles de régression sont, pour chaque algue, très différents les uns des autres. Aucun des modèles ne sélectionnent les mêmes variables. Le nombre de variables sélectionnées varie également (entre 3 pour

Algue 2 et 9 pour Algue 6). Par ailleurs on remarque que des variables sont peu sélectionnées (*Chla*, *oPO4* ne sont retenues que dans un seul modèle) tandis que d'autres se retrouvent dans quasiment tous les modèles (*mnO2* et *NH4* sont retenues dans 6 des 7 modèles).

Il n'est pas étonnant que le développement d'algues différentes répondent à des facteurs différents. Cependant, elles restent des organismes relativement similaires et le fait que certaines variables influent uniquement sur un type d'algue ou sur tous les types sauf un apparaît comme surprenant (le cas de l'algue 2 est particulièrement frappant). En effet, il n'est pas impossible que certains modèles se retrouvent avec des corrélations fortuites ou soient l'objet de surapprentissage (voir paragraphe suivant). De plus, la régression linéaire multiple ne prend en compte que les effets linéaires des variables explicatives sur la variable à prédire. Certains effets sont donc peut-être présents mais non pris en compte lors du calcul de l'estimateur du maximum de vraisemblance (qui est le même que l'estimateur des moindres carrés dans la régression linéaire), et donc ainsi dans le calcul de l'AIC pour choisir de supprimer ou non une variable. Malheureusement, nous ne disposons pas de suffisamment de connaissances au sujet des algues pour trancher.

Analyse des erreurs : Malgré des résultats assez différents pour chaque modèle, on peut en commenter certains aspects. Pour chaque modèle considéré, les erreurs de prédiction sur la base test sont toujours supérieures (voire le double) aux erreurs sur la base d'apprentissage (train), et ce quel que soit le critère d'erreur retenu (les résultats complets sont disponibles en 3.1). Le fait d'avoir des résultats prédictifs nettement meilleurs sur la base d'apprentissage que sur la base test est caractéristique d'un phénomène de **surapprentissage**. Ce phénomène semble accentué sur les modèles d'arbre de régression. Ainsi, sur les bases d'apprentissage, les modèles à arbre de décisions semblent à première vue meilleurs que les régressions linéaires mais sont en réalité plus mauvais sur les bases de test (à l'exception des Algues 6 et 7 où les résultats ne sont que légérement supérieurs sur la base de test).

On remarque également que de nombreux modèles ont une erreur NMSE et / ou NMAE supérieure à 1 : cela signifie qu'ils sont moins bons que le modèle nul, consistant à prédire la fréquence d'une algue par la moyenne observée sur la base d'apprentissage ! Ceci confirme d'autant plus la présence de surapprentissage.

En conlusion, aucun des modèles n'a de "bon" résultat prédictif (ce qui se confirme en traçant les erreurs sur un graphique). La plupart sont en effet moins bons que le modèle nul. Seules les Algues 1 et 6 se distinguent légèrement des autres avec des résultats un peu meilleurs (NMSE respectivement inférieur à 0.7 et 0.8).

3 Annexe

3.1 Résultats complets pour les variables $a2$ à $a7$

Table 4 – Résultats pour la variable a2

Prédiction de a2	MAE	MSE	RMSE	MAPE	NMSE	NMAE
Arbre de décision sur base d'apprentissage	4.9324139	61.8228962	7.8627537	Inf	0.5108776	0.6262142
Régression linéaire sur base d'apprentissage	6.2861718	95.2471	9.7594621	Inf	0.7870807	0.7980859
Arbre de décision sur base test	7.0854747	117.71136	10.8494866	Inf	1.0963262	0.9255206
Régression lineaire sur base test	6.8571224	102.668723	10.1325576	Inf	0.9562238	0.8956927

Table 5 – Résultats pour la variable a3

Prédiction de a3	MAE	MSE	RMSE	MAPE	NMSE	NMAE
Arbre de décision sur base d'apprentissage	3.1692578	26.2789922	5.1263039	Inf	0.5470145	0.6628908
Régression linéaire sur base d'apprentissage	4.3050579	40.1208972	6.3341059	Inf	0.8351429	0.9004579
Arbre de décision sur base test	4.3901411	40.6447811	6.375326	Inf	1.2867263	0.9772886
Régression lineaire sur base test	3.8578505	28.2747627	5.3174019	Inf	0.8951181	0.8587955

Table 6 – Résultats pour la variable a4

Prédiction de a4	MAE	MSE	RMSE	MAPE	NMSE	NMAE
Arbre de décision sur base d'apprentissage	1.8916822	14.5334605	3.8122776	Inf	0.7485342	0.8071263
Régression linéaire sur base d'apprentissage	2.1393638	13.1859474	3.631246	Inf	0.6791316	0.912805
Arbre de décision sur base test	1.7969618	8.4343098	2.9041883	Inf	1.0743646	0.9081953
Régression lineaire sur base test	1.8803466	7.6807532	2.7714172	Inf	0.9783764	0.9503385

Table 7 – Résultats pour la variable a5

Prédiction de a5	MAE	MSE	RMSE	MAPE	NMSE	NMAE
Arbre de décision sur base d'apprentissage	3.2378586	27.1247215	5.2081399	Inf	0.4857536	0.6112391
Régression linéaire sur base d'apprentissage	4.275731	40.749433	6.3835283	Inf	0.7297471	0.8071674
Arbre de décision sur base test	5.2334536	90.2131493	9.4980603	Inf	0.9769874	0.8571152
Régression lineaire sur base test	5.4354391	80.6201197	8.9788707	Inf	0.8730971	0.8901956

Table 8 – Résultats pour la variable a6

Prédiction de a6	MAE	MSE	RMSE	MAPE	NMSE	NMAE
Arbre de décision sur base d'apprentissage	5.5595966	85.2605484	9.2336639	Inf	0.6301951	0.7465392
Régression linéaire sur base d'apprentissage	5.8104617	81.7736386	9.0428778	Inf	0.604422	0.7802252
Arbre de décision sur base test	6.8209822	140.31086	11.8452885	Inf	0.782113	0.8050039
Régression lineaire sur base test	7.2199867	145.601005	12.0665241	Inf	0.811601	0.8520939

3.2 Code pour générer les prédictions des variables $a2$ à $a7$

```
#Même processus pour les autres variables
#NB: les NA ont déjà été remplacés dans les bases train et test
for (i in 2:7){
  #Entraîner les modèles
  algae_i = algae[,c(1:11,11+i)]
  y <- paste("a", i, sep ='')
  x <- names(algae_i)[!names(algae_i) %in% y]
  formula = as.formula(paste(y, paste(x, collapse="+"), sep="~"))
  lm.ai <- lm(formula, data = algae_i)
```

TABLE 9 – Résultats pour la variable a7

Prédiction de a7	MAE	MSE	RMSE	MAPE	NMSE	NMAE
Arbre de décision sur base d'apprentissage	2.429809	17.3115417	4.1607141	Inf	0.6538152	0.8383841
Régression linéaire sur base d'apprentissage	2.9546233	23.8603576	4.8847065	Inf	0.9011482	1.0194666
Arbre de décision sur base test	2.4819099	22.6205324	4.7561047	Inf	1.0452136	0.9228783
Régression lineaire sur base test	2.902611	24.0044	4.899428	Inf	1.109157	1.079313

```
  final.lm.ai = step(lm.ai)
  rt.ai = rpart(formula, data = algae_i)

  #Qualité de prévision sur la base train
  lm.predictions.ai = predict(final.lm.ai, algae_i)
  rt.predictions.ai = predict(rt.ai, algae_i)

  #Qualité sur la base test
  lm.predictions.ai_test = predict(final.lm.ai, test.algae)
  rt.predictions.ai_test = predict(rt.ai, test.algae)

  sink(paste("C:/Users/roman/Desktop/ENSAE 3A/Apprentissage statistique/output", paste(i, ".txt", sep=
  print('Resultats Arbre de decision sur base train')
  print(regr.eval(algae_i[[y]], rt.predictions.ai, train.y = algae_i[[y]]))
  print('Resultats Regression lineaire sur base train')
  print(regr.eval(algae_i[[y]], lm.predictions.ai, train.y = algae_i[[y]]))
  print('Resultats Arbre de decision sur base test')
  print(regr.eval(algae.sols[[y]], rt.predictions.ai_test, train.y = algae_i[[y]]))
  print('Resultats Regression lineaire sur base test')
  print(regr.eval(algae.sols[[y]], lm.predictions.ai_test, train.y = algae_i[[y]]))
  sink()

  par(mfrow = c(1, 2), col="navy", bg="lemonchiffon1")
  plot(lm.predictions.ai, algae[[y]], main = "Linear Model", xlab = "Predictions", ylab = "True Values
  abline(0, 1, lty = 2)
  plot(rt.predictions.ai, algae[[y]], main = "Regression Tree", xlab = "Predictions", ylab = "True Val
  abline(0, 1, lty = 2)

  rm(algae_i, lm.predictions.ai,rt.predictions.ai, lm.predictions.ai_test,rt.predictions.ai_test)
}
```

3.3 Traitement des données manquantes

Remarque sur la méthode knnImputation Lorsqu'on utilise la commande knnImputation sur la base d'apprentissage, 2 possibilités s'offre à nous :

— On peut exploiter l'ensemble de l'information présente sur la base d'apprentissage, c'est-à-dire y compris les variables à prédire, pour trouver les k plus proches voisins de l'observation posant problème. Cela a l'avantage de remplacer les valeurs manquantes par une valeur probablement plus proche de la "réalité", puisqu'on tient compte de tout ce que l'on sait sur l'observation. Ainsi, le modèle sera en un sens mieux entraîné, car il le sera à partir des valeurs les plus "probables" possibles. C'est ce qui est fait dans le TD avec la commande :

```
algae = knnImputation(algae, k =10, meth = "median")
```

L'inconvénient de cette méthode est de n'être pas parfaitement réplicable ensuite, lorsque le modèle est mis en production, puisqu'aucune information n'est disponible sur a priori sur les variables à prédire $a1, a2..., a7$.

— On peut donc aussi se restreindre à l'exploitation des variables explicatives. Cette démarche est plus rigoureuse, notamment si l'on raisonne ensuite par validation croisée pour sélectionner son modèle. Les résultats sur les sous-bases de test constituées sont ainsi entièrement comparables à ce que l'on aurait obtenu avec les mêmes observations en production. La commande R correspondante est :

```
algae[,1:11] = knnImputation(algae[,1:11], k =10, meth = "median")
```

3.4 Graphiques

FIGURE 1 – Arbre de décision pour l'Algue 1

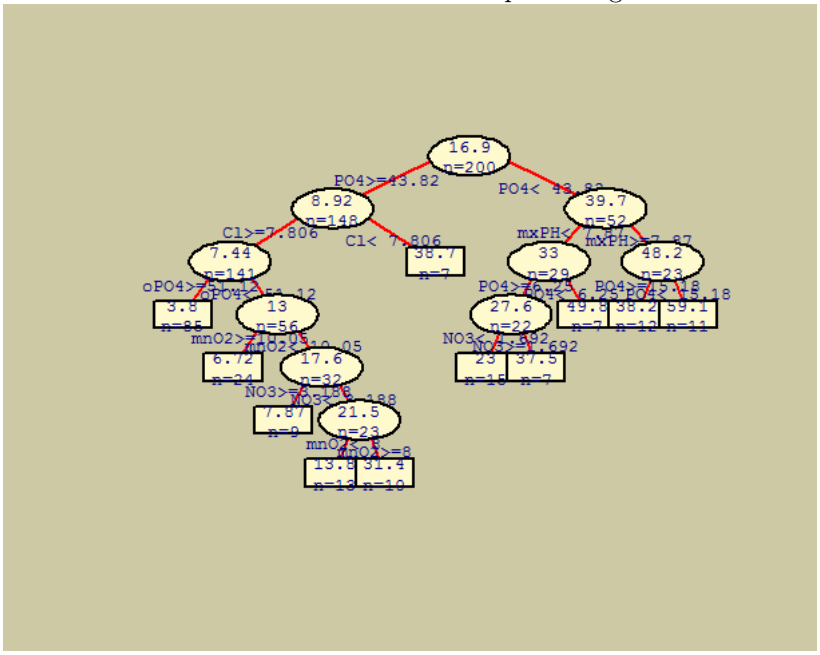

FIGURE 2 – Erreurs de prédictions sur la base test pour l'Algue 1