BOUNAB ABdelmounaim QST/REP projet

1- Précisez la nature des différentes variables. Il est nécessaire d'en étudier la distribution. Notez la symétrie ou non de celles-ci.

1. Les variables : O3obs,MOCAGE,TEMPE,RMH2O,NO2,NO,VentMOD,VentANG sont quantatives et les variables : JOUR,STATION sont qualitatives

2. La variable TEMPE est symétrique avec une distribution normale , les autre variables sont assymétriques , pour les variable qualitatives , la variable STATION est de distribution uniforme , la variable JOUR n'est pas uniforme

2- Vérifiez l'opportunité de ces transformations puis retirez les variables initiales

La transformation a rendre la variable SRMH20 plus symetrique et plus gaussienne, les deux autres variables reste toujour assymetriques.

3- Que dire sur les relations des variables 2 à 2 ?

Nous avons (LNO2,LNO) qui sont fortement corrélés, (TEMPE,O3bs),(MOCAGE,O3bs) qui ont une corélation moyenne avec (corr~0.6), les autre variable ont une corélation faible

4- Complétez en visualisant les corrélations avec la fonction corrplot() (package corrplot). Quelle est la limite de ce type de diagnostic numérique : quel type de corrélation est mesuré ?

La limite est que la corrélation est entre 2 variables uniquement.

5- Que représentent ces différents graphiques?

6- ces graphes represente le cercle de correlation entre les variables, et le nuage des individues dans les differents dimensions

6-Question Que dire du choix du nombre de dimensions, des valeurs atypiques?

le choix du nombre de dimensions est bon, car on a une somme de 72%, mais on peut ajouter d'autre dimensions pour avoir plus d'information. Pour les valeurs atypiques, nous avons pas beaucoup de valeurs atypiques

7- Que dire de la structure de corrélation des variables ? Est-elle intuitive ?

7- On peut voire que LNO2 et LNO sont fortement corélés comme les deux variables sont proche de 1 et dans la meme direction , et les variables VentMOD et ventANG sont inversé. On peut dire que la structure de corrélation est intuitive

8-Une discrimination linéaire (hyperplan) semble-t-elle possible ?

la discrimination linéaire ne semble pas possible comme les deux classes sont dificile a séparés

9-Comment appelle-t-on cette procédure spécifique de validation croisée ?

La procédure spécifique de validation croisée décrite ici est appelée "validation croisée k-fold". C'est une technique de validation croisée dans laquelle l'échantillon d'apprentissage est divisé en k souséchantillons (ou "folds"). Le modèle est ensuite ajusté k fois, chaque fois en utilisant k-1 souséchantillons comme ensemble d'apprentissage et le sous-échantillon restant comme ensemble de validation. Cette procédure permet d'estimer l'erreur de généralisation du modèle de manière plus robuste que si l'on utilisait un seul découpage apprentissage/test.

différentes méthodes sont comparées et où l'optimisation des modèles est effectuée, la validation croisée k-fold est utilisée pour évaluer les performances de chaque méthode de modélisation. Cela contribue à éviter le surajustement (overfitting) aux données d'apprentissage et fournit une estimation plus fiable de la performance des modèles sur de nouvelles données (représentées par l'échantillon test).

La notation k dans "k-fold" représente le nombre de sous-échantillons utilisés, et il est généralement choisi en fonction de la taille de l'échantillon d'apprentissage. Des valeurs courantes pour k sont 5 ou 10, mais cela peut varier en fonction des circonstances spécifiques.

En résumé, la validation croisée k-fold est une étape essentielle pour évaluer de manière robuste et impartiale les performances des différentes méthodes de modélisation dans un contexte d'apprentissage supervisé.

10- Que dire de la distribution de ces résidus ?

On peut dire que les prediction ont un grand residue (high bias), donc les valeurs prédits ne represente pas bien les valeurs actuels, on peut dire qu'il ya un under-fitting

Q11- La forme du nuage renseigne sur les hypothèses de linéarité du modèle et d'homoscédasticité.

Que dire de la validité de ce modèle ?

on peut voir que les résidus sont dispersé, donc il n'y a pas d'homoscédasticité

Q12- Ce premier modèle est comparé avec celui de la seule prévision déterministe MOCAGE. Qu'en conclure?

Ce modele donne des meilleures resultats par rapport a la prévision déterministe

Q13- Que fait la commande model.matrix ? Comment sont gérées les variables catégorielles ?

la commande model.matrix transforme les spécifications du modèle en une matrice appropriée pour l'ajustement du modèle statistique

Q14- Que représentent les courbes ci-dessus, appelées "chemins de régularisation"?

Les chemins de régularisation tracent l'évolution des coefficients des variables en fonction des valeurs de log lambda, visant ainsi à minimiser l'erreur de prédiction.

Q15- Que représente les points gras ? Et la bande qui est autour ?

les points gras represente la valeur les residues calculé avec la fonction MSE, et la bande represente l'intervalle de confiance obtenu des les differents itérations

Q16- comment sont obtenues les valeurs de log(lambda) correspondant aux lignes verticales en pointillé ?

les valeurs correspondant aux lignes sont obtenues par la valeur d'eureur minimum de MSE pour la premier ligne, pour la deuxieme ligne , elle represente la plus grand valeur de log lambda pour la quelle toutes les valeurs inferieur de lambda ont un ereur MSE inferieur a cette valeur.

Q17- Combien restent-ils de coefficients non nuls. Vérifiez sur les chemins de régularisation.

il reste que 3 non nuls

Q18- Même question en choisissant l'autre valeur de lambda retenue par glmnet, i.e. reg.lasso.cv$lambda.min

pour l'autre valeur de lambda, il ya uniquement 3 valeurs nul , il reste 10 coefficients non nuls sur 13

Q19-Commentez, Calculez le critère MSE (moyenne des carrés des résidus) pour les deux modèles. Pourquoi celu i obtenu par LASSO est-il moins bon ? Quel critère LASSO minimise t-il ?

les modeles ne sont pas trés dispersé, donc on peut dire que les modeles ont un bon précision dans la prédiction, et on peut voire que le model avec lambda min est mieux que lambda 1se, le modele sans selection est le meilleur

2. la methode avec LASSO a un MSE plus grand que le modéle sans sélection , la methode LASSO minimise le MSE en ajoutant une pénalité de norme L1, encourageant la sparsité des coefficients et la sélection de variables.

Q20- Estimez l'erreur du modèle linéaire simple sans sélection de variables par validation croisée. Comparez avec celle du LASSO. Qu'observez-vous?

J'ai observé que le modèle LASSO min a une erreur de prédiction plus faible par rapport au modele sans selection , mais LASSO 1se est pire que le modele sans selection , donc la methode lasso peut donner des meilleur resultats quand on choisit le meilleur lambda

Q21- Quel autre critère, équivalent à AIC dans le cas gaussien et de variance résiduelle connue, est utilisé en régression linéaire ?

l'autre critére est le critère de Mallows

Q22- Que sont sensibilité et spécificité d'une courbe ROC?

la courbe ROC évalue la performance d'un modèle de classification à travers divers seuils de décision. sensibilité mesure l'aptitude du modèle à correctement identifier les instances positives, tandis que la spécificité mesure sa capacité à correctement identifier les instances négatives.

Q23- Les performances des deux approches gaussiennes et binomiales sont-elles très différentes ?

Les deux approches présentent des résultats de prédictions similaires en termes d'erreurs, donc les modeles sont pas trés différentes

Q24- Sur le graphe ci-dessus, ajouter la courbe ROC pour le modèle déterministe MOCAGE. Qu'observez-vous?

On observe que le graph de MOCAGE est pire que les deux autre

Q25-Que sont sensibilité et spécificité d'une courbe ROC?

le critére optimisé est la somme des carrés des résidus

Q26- Quelle est la variable qui contribue le plus à l'interprétation ?

la variable est MOCAGE

Q27- A quoi est due la structure particulière de ce graphe ?

la structure de graph provient de la nature binaire du modèle d'arbre, où chaque prédiction particulière est représentée par les feuilles de l'arbre (par exemple ici 9 valeurs represente 9 feuilles)

Q28- Quel autre critère d'hétérogénéité est utilisé ?

Le gain d'information ou Gini

Q27- Comparez avec les erreurs précédentes estimées également par validation croisée. Quelle analyse discriminante retenir ? Pourquoi ?

La comparaison des erreurs estimées par validation croisée permet de choisir la méthode discriminante avec la meilleure performance prédictive. Choisissez la méthode avec l'erreur la plus faible.

Q28- Une méthode est-elle uniformément meilleure sur cet échantillon test ?

La courbe ROC illustre la performance des méthodes sur l'échantillon test en termes de trade-off entre taux de faux positifs et vrais positifs. Pour déterminer si une méthode est uniformément meilleure, examinez l'aire sous la courbe (AUC). Une AUC plus élevée indique une meilleure performance globale.

Q29- Le temps d'exécution pour les SVM est-il plus sensible au nombre d'observations ou au nombre de variables ? Pourquoi ?

Le temps d'exécution pour les SVM (Support Vector Machines) est généralement plus sensible au nombre d'observations (échantillon) qu'au nombre de variables (dimensions). Cela est dû au fait que le coût computationnel des SVM dépend du nombre d'échantillons dans l'ensemble d'apprentissage.

Les SVM, en particulier les SVM à noyau, construisent une frontière de décision en se basant sur un sous-ensemble de vecteurs de support, qui sont les échantillons les plus influents pour déterminer la séparation entre les classes. Plus le nombre d'échantillons est élevé, plus le calcul des vecteurs de support et des coefficients associés devient coûteux.

En revanche, le nombre de variables a un impact sur le temps d'exécution, mais généralement de manière moins significative. Les SVM sont efficaces pour gérer un grand nombre de dimensions (variables) en raison de leur capacité à fonctionner dans des espaces de grande dimension.

Q30- Notez la pénalisation optimale pour le noyau considéré (Gaussien). Ré-estimez le modèle supposé optimal avant de tracer le graphe des résidus. Comme précédemment, observez que plusieurs exécutions conduisent à des résultats différents et donc que l'optimisation de ce paramètre est pour le moins délicate.

Quels autres noyaux sont disponibles dans cette implémentation des SVM ?

Les résultats des paramètres optimaux peuvent varier en raison de l'échantillonnage aléatoire dans le processus de validation croisée.

Quant aux noyaux disponibles dans cette implémentation des SVM, les noyaux les plus couramment utilisés sont :

1. Linéaire : kernel = "linear"
2. Polynomiale : kernel = "polynomial"
3. 4. Radial (Gaussien) : kernel = "radial"

   Sigmoid : kernel = "sigmoid"

On peut spécifier le noyau désiré en utilisant le paramètre kernel dans la fonction svm .

Q31- Qu'est-ce qui cause le rapprochement des résidus dans un "couloir"? Qu'observez-vous lorsque vous faites varier les paramètres cost et epsilon?

Le rapprochement des résidus dans un "couloir" peut être causé par une forte pénalisation (paramètre cost élevé) ou une marge plus étroite (paramètre epsilon faible) dans le contexte des SVM. Ce phénomène est souvent observé lors de l'utilisation de SVM avec une marge souple.

- Pénalisation élevée (cost élevé) : Lorsque le paramètre cost est élevé, le modèle SVM est fortement assé. Cela conduit à une optimisation qui accorde une importance élevée à chaque observation, ce qui peut entraîner un ajustement trop serré aux données d'apprentissage, formant ainsi un "couloir" étroit entre les deux

  pénalisé pour chaque point situé à l'intérieur de la marge ou mal cl

classes.

- Marge étroite (epsilon faible) : La largeur de la marge dans les SVM est contrôlée par le paramètre est faible, la marge est étroite, et les observations qui se trouvent à

l'intérieur de la marge ou du mauvais côté de la frontière de décision ont un impact plus important sur

epsilon. Lorsque epsilon le modèle. Cela peut également conduire à un "couloir" étroit.

En faisant varier ces paramètres, vous pouvez observer des changements dans la largeur du "couloir" et la manière dont le modèle réagit aux points aberrants ou aux observations mal classées. Il s'agit d'un compromis délicat entre la complexité du modèle, la marge, et la tolérance aux erreurs d'apprentissage. Un ajustement trop serré peut conduire à un surajustement aux données d'apprentissage, tandis qu'un ajustement trop lâche peut entraîner une sous-estimation du modèle.

Q32- Les SVM apportent-ils une amélioration?

L'efficacité des SVM dépend du problème et du réglage des paramètres. Leur utilité doit être évaluée par comparaison méthodique.

Q33- Quel critère est optimisé lors de la création d'un noeud de l'arbre?

2 /zzLors de la création d'un nœud de l'arbre de régression, le critère optimisé est généralement l'erreur quadratique moyenne (Mean Squared Error - MSE).

Q34- A quoi est due la structure particulière de ce graphe ?

La structure particulière du graphe des résidus de l'arbre de régression peut être due au fait que l'arbre divise l'espace des prévisions en segments constants, ce qui entraîne des résidus groupés. Ces groupes correspondent aux feuilles de l'arbre, et chaque feuille prédit une valeur constante pour les observations qui y aboutissent.

Q35- Question Quel autre critère d'hétérogénéité est utilisé ?

Outre l'indice de concentration de Gini, un autre critère d'hétérogénéité utilisé dans l'algorithme CART (Classification and Regression Trees) est l'entropie, également appelée "information gain".

Q35- Comparez les qualités de prévision. Une meilleure méthode se dégage-t-elle ?

Les courbes ROC permettent de comparer visuellement les performances des modèles de régression logistique (Logit), de l'arbre de régression (TreeReg), et de l'arbre de classification (TreeDis). La comparaison se fait en termes de taux de vrais positifs (tpr) et de taux de faux positifs (fpr) à différents seuils de probabilité.

La courbe ROC de la régression logistique (Logit) est représentée en bleu, celle de l'arbre de régression (TreeReg) en orange avec un style de ligne en pointillé, et celle de l'arbre de classification (TreeDis) en vert.

En comparant ces courbes, vous pouvez évaluer la capacité des modèles à discriminer entre les classes positives et négatives. Une courbe qui se rapproche davantage du coin supérieur gauche (avec un tpr élevé et un fpr faible) indique une meilleure performance. La légende à droite indique les couleurs correspondant à chaque modèle.

Il semble que la régression logistique (Logit) ait une meilleure performance sur la base de la courbe ROC, mais une évaluation plus détaillée des métriques de performance et de la nature spécifique du problème pourrait être nécessaire pour confirmer cette conclusion.

la methode de regression Logit est clairement la meilleure , on peut voir aussi que la methode de l'arbre de discrimination est mieux l'arbre de regression TreeReg

Q36- Quel est le paramètre mtry de la fonction randomForest?

Le paramètre mtry dans la fonction randomForest spécifie le nombre de variables à sélectionner aléatoirement à chaque division d'un nœud de l'arbre. En d'autres termes, il contrôle le nombre de prédicteurs (variables) considérés à chaque étape lors de la construction de chaque arbre de la forêt aléatoire.

Un choix judicieux de mtry peut contribuer à la diversité des arbres dans la forêt, ce qui est souvent bénéfique pour les performances globales du modèle. Une valeur communément recommandée pour mtry est la racine carrée du nombre total de prédicteurs dans les données.

La diversité accrue des arbres contribue à la robustesse du modèle, car chaque arbre est formé sur un sous-ensemble différent de prédicteurs, ce qui peut aider à éviter le surajustement (overfitting). Cependant, la valeur optimale de mtry peut dépendre du problème spécifique et peut nécessiter une exploration via une validation croisée ou une autre technique d'optimisation.

Q37- En quoi le bagging est un cas particulier des forêts aléatoires ? Le bagging ne sera pas traité dans ce TP.

Le bagging (Bootstrap Aggregating) est une technique d'ensemble qui consiste à construire plusieurs modèles indépendants en utilisant des échantillons bootstrap de l'ensemble de données d'entraînement, puis à agréger leurs prédictions. L'idée principale est d'introduire de la diversité en ajustant chaque modèle sur un sous-ensemble différent des données, ce qui contribue à réduire la variance globale du modèle.

Les forêts aléatoires sont une extension du bagging appliquée spécifiquement aux arbres de décision. En plus de l'échantillonnage bootstrap sur les observations, les forêts aléatoires introduisent une autre source de diversité en sélectionnant aléatoirement un sous-ensemble de prédicteurs (variables) à chaque division de chaque arbre. Cela se fait en fixant le nombre de prédicteurs à considérer (mtry), comme mentionné précédemment.

Ainsi, on peut considérer le bagging comme un cas particulier des forêts aléatoires lorsque la sélection aléatoire de prédicteurs n'est pas utilisée, c'est-à-dire lorsque mtry est égal au nombre total de prédicteurs. Les forêts aléatoires étendent le concept de bagging en introduisant davantage de diversité grâce à la sélection aléatoire de prédicteurs, ce qui peut améliorer les performances dans de nombreux cas.

Q38- Quelles est la valeur par défaut de mtry ?

La valeur par défaut de mtry dans la fonction randomForest de R est généralement la racine carrée du nombre total de prédicteurs (variables). Cela signifie que lors de la construction de chaque arbre de la forêt aléatoire, un sous-ensemble aléatoire de prédicteurs est considéré à chaque division, avec le nombre de prédicteurs égal à la racine carrée du nombre total de prédicteurs.

En d'autres termes, si vous avez p prédicteurs, la valeur par défaut de mtry serait généralement racine p. Cela permet d'introduire de la diversité dans chaque arbre en sélectionnant un sous-ensemble aléatoire de prédicteurs, contribuant ainsi à la puissance de généralisation de la forêt aléatoire.

Q39- Quelle est la valeur par défaut de mtry?

La valeur par défaut de mtry dans randomForest de R est la racine carrée du nombre total de prédicteurs.

Q40- Commentez les erreurs, testez d'autres exécutions avec d'autres valeurs des paramètres.

Il semble y avoir des problèmes d'inversion des classes dans le modèle, comme indiqué par la faible performance de la classification des deux classes sur les données d'apprentissage (OOB). Il peut être utile de réexécuter le modèle avec différentes valeurs de paramètres, notamment en ajustant le nombre d'arbres (ntree), la profondeur maximale des arbres (maxdepth), ou d'autres paramètres pertinents. L'interprétation des résultats de l'importance des variables (importance) peut également aider à comprendre quelles variables ont un impact significatif sur la prédiction du modèle.

Q41- Question Quelles sont les deux mesures d'importance des variables ?

Les deux mesures d'importance des variables présentées dans la sortie sont les suivantes :

1. MeanDecreaseAccuracy : Il mesure la diminution de la précision moyenne du modèle lorsque chaque

variable est exclue. Une plus grande diminution de la précision indique une variable plus importante.

2. MeanDecreaseGini : Il mesure la diminution de l'indice de Gini moyenné sur toutes les divisions du modèle lorsque chaque variable est exclue. Une plus grande diminution de l'indice de Gini suggère une variable plus importante pour la séparation des classes.

Ces mesures aident à évaluer l'importance relative des variables dans la construction du modèle.

Q42- Qu'indique la comparaison des courbes ROC ?

La comparaison des courbes ROC fournit une évaluation visuelle des performances des différents modèles en termes de trade-off entre la sensibilité et la spécificité. Voici quelques interprétations possibles en fonction des caractéristiques des courbes ROC :

1. Position vers le coin supérieur gauche : Une courbe ROC qui s'approche du coin supérieur gauche

indique une meilleure performance, car elle signifie une plus grande sensibilité (taux de vrais positifs) et une plus grande spécificité (taux de vrais négatifs).

2. Courbe vers le coin supérieur gauche : Une courbe qui se rapproche davantage du coin supérieur gauche suggère une meilleure discrimination entre les classes (dans ce cas, dépassement du seuil ou non).

3. Superposition des courbes : Si deux courbes ROC se chevauchent, l'interprétation dépend du contexte. Si l'une est légèrement au-dessus de l'autre sur toute la plage, cela peut indiquer une meilleure performance globale. Cependant, il peut également être utile de comparer les aires sous les courbes (AUC) pour une évaluation quantitative.

Q43- Comment intervient le shrinkage en boosting?

Le shrinkage, également appelé learning rate, est un paramètre important dans les méthodes de boosting, notamment dans le cadre des algorithmes comme le Gradient Boosting. Le shrinkage contrôle la contribution de chaque arbre au modèle final. Voici comment il intervient dans le processus de boosting :

1. Apprentissage progressif : Le boosting construit un modèle de prédiction en ajoutant itérativement

des arbres faibles au modèle existant. À chaque itération, le modèle apprend à corriger les erreurs résiduelles du modèle précédent.

2. Shrinkage : Le shrinkage réduit l'impact de chaque nouvel arbre ajouté au modèle. Il est

généralement un petit nombre entre 0 et 1. Lorsque le shrinkage est proche de 1, chaque nouvel arbre

a un impact important sur la correction des erreurs. En revanche, lorsque le shrinkage est plus petit, chaque arbre contribue de manière moins significative.

3. Contrôle de surajustement : Un faible shrinkage aide à contrôler le surajustement. En utilisant un shrinkage plus petit, le modèle final dépend davantage de l'ensemble des arbres, ce qui peut améliorer la généralisation du modèle.

4. Nombre d'itérations : En raison du shrinkage, il peut être nécessaire d'ajouter davantage d'itérations (arbres) pour atteindre une performance équivalente à celle d'un modèle avec un shrinkage plus élevé.

Le choix du shrinkage est un compromis entre la vitesse de convergence (plus petit shrinkage nécessitant plus d'itérations) et la régularisation du modèle (plus petit shrinkage réduisant le surajustement). Il est souvent ajusté via une procédure de validation croisée pour trouver la meilleure valeur dans le contexte spécifique du problème.

Q44- Pour quel boosting? Ou que signifie gbm?

GBM (Gradient Boosting Machine) est une technique d'apprentissage automatique basée sur l'idée de construire un modèle prédictif en ajoutant des arbres de décision faibles itérativement. Le terme "Gradient Boosting" fait référence à l'utilisation d'une descente de gradient pour minimiser la fonction de perte du modèle, en ajustant progressivement le modèle pour réduire les erreurs résiduelles.

GBM est un algorithme de boosting qui peut être utilisé pour des tâches de régression et de classification. Il est connu pour sa flexibilité, sa capacité à gérer des données hétérogènes, et son efficacité en termes de performances prédictives.

Voici quelques points clés à retenir sur GBM :

1. Boosting : GBM est une méthode de boosting, où chaque nouvel arbre est construit pour corriger les erreurs résiduelles du modèle existant.

2. Gradient Descent : La descente de gradient est utilisée pour minimiser la fonction de perte en ajustant les paramètres du modèle de manière itérative.

3. Arbres Faibles : Chaque arbre ajouté au modèle est un arbre faible, souvent appelé "stump" (tronc). Ces arbres sont simples et peuvent être des arbres de profondeur limitée.

4. Ensemble d'arbres : Le modèle final est un ensemble (ou une combinaison) d'arbres, où chaque arbre contribue à la prédiction finale.

GBM peut être mis en œuvre avec différents logiciels et bibliothèques, et il existe plusieurs implémentations bien connues, notamment l'implémentation en R appelée "gbm" et celle en Python appelée "scikit-learn".

Q45- Quelle stratégie d'agrégation de modèles vous semble fournir le meilleur résultat de prévision?

Le choix de la meilleure stratégie d'agrégation de modèles dépend souvent de la nature des données et du problème spécifique à résoudre. Deux des stratégies d'agrégation de modèles les plus couramment utilisées sont le Bagging (Bootstrap Aggregating) et le Boosting. Chacune de ces stratégies a ses propres caractéristiques et peut être plus appropriée dans certaines situations.

1. Bagging (Bootstrap Aggregating) :

Principe : Bagging consiste à construire plusieurs modèles indépendants en utilisant des mbles aléatoires de données (échantillonnage bootstrap) et à agréger leurs s.

: Il réduit la variance en moyennant les prédictions de modèles divers. Il est moins e de surajuster les données d'entraînement.

Random Forest est une méthode basée sur le bagging qui utilise des arbres de

omme modèles de base.

sous-ense prédiction Avantages susceptibl Exemple : décision c

2. Boosting :

-

-

- Principe : Boosting construit une séquence de modèles où chaque modèle tente de corriger s résiduelles du modèle précédent. Les modèles sont agrégés pondérés.

: Il peut améliorer la précision du modèle même avec des modèles faibles. Il est our réduire le biais et améliorer la prédiction.

Gradient Boosting Machine (GBM) est une méthode de boosting populaire.

les erreur

Avantages

efficace p

Exemple :

Le choix entre Bagging et Boosting dépend de plusieurs facteurs, tels que la complexité du problème, la taille de l'ensemble de données, la diversité des modèles de base, et le compromis entre biais et variance. Dans certains cas, une combinaison de ces stratégies, comme dans le cas de Random Forest qui utilise le bagging et le boosting, peut également être efficace. Il est généralement recommandé d'expérimenter différentes approches pour déterminer celle qui fonctionne le mieux pour une tâche spécifique.

Q46- Est-elle, sur ce jeu de données, plus efficace que les modèles classiques expérimentés auparavant ?

Les performances d'une méthode d'agrégation de modèles, telle que Random Forest, par rapport aux modèles classiques dépendent de divers facteurs, notamment la nature des données, la qualité de l'ajustement du modèle, la diversité des modèles de base, etc. Pour évaluer si Random Forest est plus efficace que les modèles classiques expérimentés précédemment, vous devriez effectuer une comparaison des performances.

Voici quelques étapes que vous pourriez suivre pour comparer les performances :

1. 2. Divisez vos données : Divisez votre ensemble de données en ensembles d'apprentissage et de test.

Entraînez les modèles : Entraînez Random Forest ainsi que les modèles classiques sur l'ensemble

d'apprentissage.

3. Évaluez les performances : Utilisez l'ensemble de test pour évaluer les performances de chaque modèle en utilisant des métriques appropriées telles que la précision, le rappel, la courbe ROC, etc.

4. Comparez les résultats : Comparez les performances des différents modèles pour déterminer celui qui offre les meilleures performances sur votre ensemble de données spécifique.

Q47- Quel est le paramètre decay de la fonction nnet?

Dans la fonction nnet du package R, le paramètre decay contrôle le terme de régularisation pour éviter le sur-apprentissage (overfitting) dans les réseaux de neurones. Le terme de régularisation aide à prévenir le sur-apprentissage en ajoutant une pénalité aux poids du réseau de neurones. Une valeur de

decay plus élevée entraîne une régularisation plus forte.

Q48- Indiquez une autre façon d'éviter le sur-apprentissage.

Une autre façon d'éviter le sur-apprentissage est d'utiliser la validation croisée (cross-validation). La validation croisée divise l'ensemble de données en plusieurs parties, entraîne le modèle sur une partie et évalue sa performance sur une autre partie. Cela permet d'obtenir une estimation plus robuste des performances du modèle, car il est évalué sur des données qu'il n'a pas vues pendant l'entraînement. Cela aide à identifier si le modèle est trop spécifique aux données d'entraînement (sur-apprentissage) ou s'il peut généraliser correctement sur de nouvelles données.

Q49- Pour chaque cas, identifiez la méthode, précisez les paramètres associés et notez celui ou ceux optimisés par défaut par caret

1. Régression logistique : Le paramètre associé est la complexité du modèle. Le paramètre optimisé par défaut est "size," qui détermine la taille du réseau de neurones.

2. Arbre de décision : Le paramètre associé est la complexité du modèle. Le paramètre optimisé par défaut est "cp," la complexité de la scission.

3. Réseau de neurones : Les paramètres associés sont la taille du réseau (size) et la fonction d'activation.

Les paramètres optimisés par défaut sont "size" et "decay."

4. Forêt aléatoire : Les paramètres associés sont le nombre de variables à tester à chaque division (mtry) et la complexité du modèle. Le paramètre optimisé par défaut est "mtry."

5. Boosting : Les paramètres associés sont le taux d'apprentissage (shrinkage) et la complexité du modèle. Les paramètres optimisés par défaut sont "shrinkage" et "n.trees."

Q50- Quelle méthode retenir, en fonction du taux de faux positifs acceptable, pour prévoir le dépassement du seuil? Et si le commanditaire veut une solution explicable?

Le choix de la méthode dépend des objectifs du commanditaire. Si le taux de faux positifs doit être minimisé, alors la méthode qui offre la courbe ROC avec le plus bas taux de faux positifs pour un taux de vrais positifs donné devrait être privilégiée. Cependant, si le commanditaire souhaite une solution explicable, la régression logistique (logit) pourrait être préférée, car elle produit des coefficients qui sont interprétables et permettent de comprendre l'impact de chaque variable sur la prédiction. Cela dépend des compromis entre la complexité du modèle, la performance et l'interprétabilité.

Q51- Quel est en moyenne le nombre de données manquantes par colonne?

Pour déterminer en moyenne le nombre de données manquantes par colonne dans la matrice Xna, vous pouvez utiliser la fonction colMeans pour calculer la proportion de valeurs manquantes (NA) dans chaque colonne.

Q52- Qu'en serait-il en utlisant Python au lieu de R ?

En Python, pour imputer les valeurs manquantes, vous pouvez utiliser des bibliothèques comme scikitlearn (moyenne, kNN), pandas (moyenne, médiane), et fancyimpute (SoftImpute, kNN). Les choix dépendent du contexte et des besoins spécifiques de l'analyse.

Q53- Quel est le rôle du paramètre k ci-dessous?

Le paramètre k dans la fonction lof de la bibliothèque Rlof est utilisé pour spécifier le nombre de voisins à considérer lors du calcul de la fonction de dissimilarité (Local Outlier Factor - LOF). Dans l'algorithme LOF, chaque point est évalué en fonction de la densité locale de ses voisins par rapport à lui-même. Le paramètre k détermine le nombre de voisins pris en compte pour mesurer cette densité locale. Un k plus élevé signifie que l'algorithme prend en compte un plus grand nombre de voisins pour évaluer la densité locale.

Dans le contexte de votre code, k=c(3:7) indique que vous souhaitez utiliser les points voisins de 3 à 7 inclusivement pour calculer les valeurs LOF. Vous pouvez ajuster cette plage en fonction de la nature de vos données et de la sensibilité aux valeurs aberrantes que vous souhaitez obtenir.

Q54- Comment intervient la borne 1.5? A quelle classe appartiennent majoritairement les observations jugées atypiques ?

La borne 1.5 est souvent utilisée dans la méthode des valeurs aberrantes (outliers) pour déterminer les points atypiques. Elle correspond à multiplier l'écart interquartile par 1.5. Les observations au-delà de cette borne sont considérées comme potentiellement atypiques. Majoritairement, les observations atypiques appartiennent à la classe des valeurs élevées.

Q55- One Class Classification SVM

Question Quel est le rôle du paramètre nu?

Le paramètre ν dans la méthode de classification en une seule classe (One-Class SVM) contrôle la fraction d'observations aberrantes (outliers) attendues dans l'ensemble d'apprentissage. Il spécifie la proportion maximale d'observations d'apprentissage qui peuvent être classées comme aberrantes. Un ν plus bas permet d'accepter un pourcentage plus élevé d'observations comme aberrantes, tandis qu'un ν plus élevé les restreint davantage. Ainsi, ν est une manière de régler le niveau de tolérance visà-vis des observations considérées comme atypiques dans le modèle One-Class SVM.

Q56- Commenter la répartition des atypiques au sens de Random Forest. Serait-il raisonnable de supprimer ces observations ?

La répartition des observations considérées comme atypiques par Random Forest peut donner des indications importantes sur la qualité et la pertinence de ces observations dans le contexte du modèle. Si un grand nombre d'observations est classé comme atypique, cela peut indiquer qu'elles sont potentiellement différentes du reste de l'ensemble de données et pourraient être des valeurs aberrantes ou des erreurs de mesure.

Cependant, la décision de supprimer ces observations dépend de plusieurs facteurs, notamment la nature des données, le contexte de l'étude et l'objectif de la modélisation. Il est important de considérer les raisons potentielles derrière le classement comme atypique. Parfois, ces observations peuvent contenir des informations importantes ou des tendances intéressantes. Dans d'autres cas, elles peuvent être des erreurs de mesure ou des valeurs aberrantes indésirables.

Q57 Que dire sur la correspondance entre les trois stratégies de détection d'observations atypiques?

Q58 -Qu'est-ce qui permettrait d'en choisir une parmi les trois ou parmi les très nombreuses autres méthodes disponibles dans la littérature?

La correspondance entre les trois stratégies de détection d'observations atypiques (LOF, isolation forest, SVM One-Class) dépend des caractéristiques spécifiques des données et des objectifs de l'analyse. Chaque méthode a ses propres avantages et limitations, et il n'y a pas de méthode unique

qui convienne à toutes les situations. Voici quelques points à considérer pour choisir entre ces méthodes ou d'autres disponibles :

1. Nature des données : Certaines méthodes peuvent être plus adaptées à des types spécifiques de

données. Par exemple, l'isolation forest est souvent efficace pour détecter des valeurs aberrantes dans des ensembles de données de grande dimension, tandis que LOF peut bien fonctionner pour des données plus complexes avec des clusters de différentes densités.

2. Interprétabilité : Certaines méthodes, comme LOF, fournissent des scores qui peuvent être interprétés comme des degrés d'atypicité. Cela peut être utile pour comprendre la gravité de l'atypicité. D'autres méthodes, comme l'isolation forest, se concentrent sur l'isolement d'observations atypiques sans fournir de scores interprétables.

3. Robustesse : La robustesse de la méthode face à différentes distributions de données et à des proportions variables d'observations atypiques peut être un critère de choix. Par exemple, certaines méthodes peuvent être plus robustes en présence de données fortement déséquilibrées.

4. Complexité et temps d'exécution : La complexité algorithmique et le temps d'exécution peuvent varier entre les méthodes. Si la rapidité d'exécution est cruciale, cela peut influencer le choix de la méthode.

5. Validation : Il est recommandé de valider la performance de chaque méthode sur des ensembles de données spécifiques à l'aide de techniques de validation croisée ou d'autres méthodes d'évaluation. Une évaluation approfondie sur des jeux de données de test peut aider à déterminer quelle méthode fonctionne le mieux dans un contexte donné.

Consclusion :

Au cours de mon étude des méthodes d'apprentissage automatique, j'ai acquis des compétences approfondies, notamment en statistiques descriptives, régression linéaire, régression logistique, analyse en composantes principales, arbres de décision et réseaux de neurones. Cette formation a considérablement enrichi mes aptitudes analytiques, me conférant une expertise robuste en apprentissage automatique. Parallèlement, elle m'a introduit de manière méthodique à l'utilisation avancée du langage R. Ces compétences me permettent maintenant de résoudre des problèmes complexes et de prendre des décisions éclairées dans des contextes pratiques.