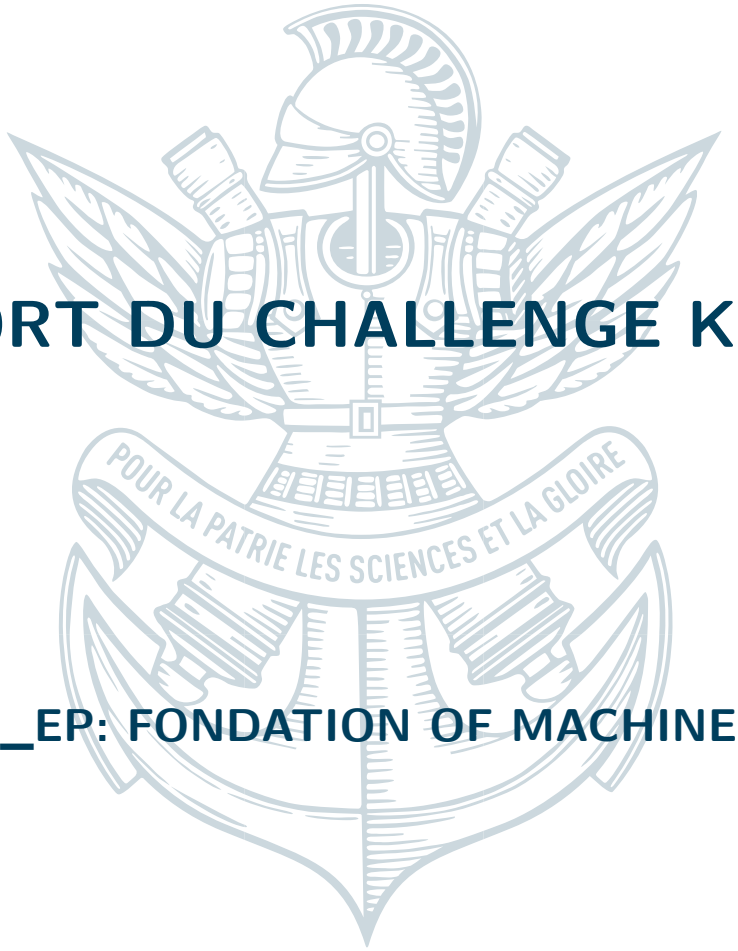

RAPPORT DU CHALLENGE KAGGLE

APM_51053_EP: FONDATION OF MACHINE LEARNING

Ouattara Aboubakar /Borel Domgue / Ruflin Tsasse

SOMMAIRE

Abstract **2**

Introduction **3**

1 Description/Sélection des données **4**

2 Modèle XGBoost **6**

3 Modèle Light GBM **8**

4 Modèle hybride **9**

5 Résultats et commentaires **10**

Conclusion **11**

Bibliographie **12**

ABSTRACT

La cartographie des types de couverture forestière revêt une importance capitale, tant du point de vue environnemental qu'économique et social. Celle-ci regroupe la compréhension des écosystèmes (notamment en régulant le climat, en soutenant la biodiversité), le suivi de la déforestation et des changements climatiques, et enfin les modélisations et prévisions environnementales. Face à la quantité d'informations forestières dont on a accès aujourd'hui, il serait donc plus facile de faire une catégorisation à partir de modèles prédictifs. Dans cette étude, nous avons utilisé deux modèles d'apprentissage automatique (XGBoost et LightGBM), combinés avec une sélection de caractéristiques statistiques, pour classifier différents types de couverture à partir de variables cartographiques. Nos résultats montrent que la combinaison de ces deux modèles permet d'obtenir les meilleures performances parmi ceux que nous avons implémentés. Ce travail souligne l'efficacité des méthodes hybrides dans l'amélioration de la précision des prédictions.

INTRODUCTION

La catégorisation des types de couverture forestière présente un intérêt majeur dans de nombreux domaines stratégiques et scientifiques. Elle permet de mieux comprendre la dynamique des écosystèmes forestiers, en identifiant les changements de végétation au fil du temps et en évaluant l'impact des variations climatiques sur ces environnements. Cette information est essentielle pour orienter les politiques de conservation de la biodiversité et pour mieux gérer les ressources naturelles, notamment les forêts, qui jouent un rôle crucial dans la séquestration du carbone et la régulation du climat. De plus, la cartographie des combustibles exploitables et l'évaluation de la disponibilité en eau des sols permettent de renforcer les stratégies de gestion des risques, comme la prévention des incendies de forêt et la planification des ressources hydriques.

Sur le plan méthodologique, l'automatisation de cette catégorisation par l'apprentissage automatique et l'imagerie satellite offre un avantage décisif en termes de rapidité, de précision et de réduction des coûts, par rapport aux méthodes traditionnelles de terrain. Ce type de catégorisation facilite également la gestion à grande échelle de vastes territoires, ce qui s'avère crucial pour les agences de gestion des terres et les acteurs du secteur forestier.

Notre objectif ici est d'évaluer et de comparer différentes méthodes d'apprentissage automatique pour classifier différents types de couverture forestière. De ce fait, nous nous sommes intéressés aux *Decision Tree*, la machine à vecteurs de support (SVM), le *random forests* (RF) et le modèle hybride XGBoost avec LightGBM.

1 DESCRIPTION/SÉLECTION DES DONNÉES

Description

Le domaine d'étude concerné par ce projet comprend les régions sauvages de Rawah, Neota, Comanche Peak et Cache la Poudre situées dans la forêt nationale de Roosevelt, au nord du Colorado (voir Figure 1).

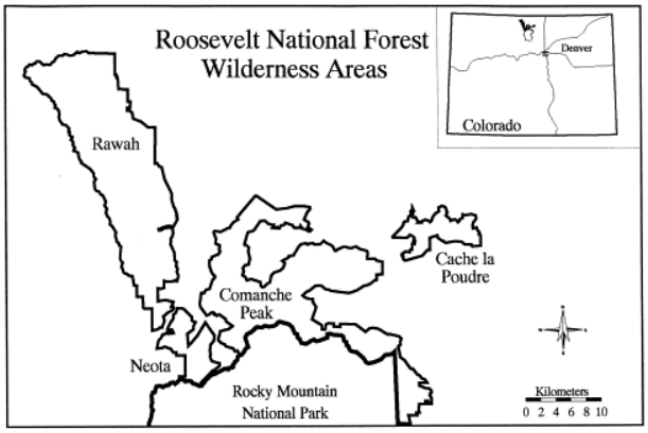

Figure 1 – map

Le choix de ces régions sauvages n'est pas un hasard, en effet ces zones contiennent des terres forestières ayant subi relativement peu de perturbations directes dues à la gestion humaine. Par conséquent, on peut donc dire que la composition actuelle des types de couvertures forestières dans ces zones résulte principalement de processus écologiques naturels, plutôt que de la gestion forestière active.

Dans notre étude, les différents modèles sus-cités ont utilisé une procédure de classification supervisée pour attribuer chaque observation à l'une des sept classes de couverture forestière mutuellement exclusives. Ces sept types de classes sont les suivantes : Spruce/Fir, Lodgepole Pine, Ponderosa Pine, Cottonwood/Willow, Aspen, Douglas-fir, Krummholz. Ces sept types de couvertures forestières représentent effectivement les principales espèces d'arbres dominantes actuellement présentes dans les quatre zones sauvages. Les cartes des types de couvertures pour ces zones ont été créées par le US Forest Service (USFS).

Pour cette étude, les données obtenues auprès du US Geological Survey (USGS) et du US Forest Service (USFS) ont été utilisées pour dériver les variables indépendantes des modèles prédictifs. Les 12 variables suivantes (avec leurs unités de mesure) ont été utilisées :

— **Elevation**, quantitative (mètres) : Élévation en mètres.

— **Aspect**, quantitative (azimut) : Aspect en degrés azimut.

— **Slope**, quantitative (degrés) : Pente en degrés.

— **Horizontal_Distance_To_Hydrology**, quantitative (mètres) : Distance horizontale à la source d'eau de surface la plus proche.

— **Vertical_Distance_To_Hydrology**, quantitative (mètres) : Distance verticale à la source d'eau de surface la plus proche.

— **Horizontal_Distance_To_Roadways**, quantitative (mètres) : Distance horizontale à la route la plus proche.

— **Hillshade_9am**, quantitative (index de 0 à 255) : Indice d'ombrage à 9h, solstice d'été.

— **Hillshade_Noon**, quantitative (index de 0 à 255) : Indice d'ombrage à midi, solstice d'été.
— **Hillshade_3pm**, quantitative (index de 0 à 255) : Indice d'ombrage à 15h, solstice d'été.
— **Horizontal_Distance_To_Fire_Points**, quantitative (mètres) : Distance horizontale au point d'ignition de feu de forêt le plus proche.
— **Wilderness_Area** (4 colonnes binaires), qualitative (0 (absence) ou 1 (présence)) : Désignation de la zone sauvage.
— **Soil_Type** (40 colonnes binaires), qualitative (0 (absence) ou 1 (présence)) : Désignation du type de sol.

Au total donc, quatre zones sauvages et 40 classes de types de sol ont été utilisées dans cette étude, produisant ainsi quatre variables de désignation des zones sauvages, quarante variables de désignation des types de sol et dix variables continues, soit un total de 54 variables indépendantes possibles disponibles pour chaque modèle. Il est également important de noter que les données fournies pour le challenge ont déjà pris en compte les variables qualitatives, telles que *soil_type* et *Wilderness_Area*, en les encodant. Cela nous a permis de disposer d'un dataset comportant 56 colonnes.

Sélection de l'ensemble des données

Dans notre travail, trois ensembles de données mutuellement exclusifs et distincts ont été créés pour entraîner, valider et tester les modèles prédictifs. D'abord, un ensemble de données d'entraînement (80%) a été utilisé pour développer les classifieurs des modèles de prédiction basés sur le modèle hybride (XGBoost et LightGBM), le *random forests* (RF) et le *Decision Tree*. L'ensemble de validation (20%) avait pour objectif de tuner les hyperparamètres du modèle et de vérifier sa capacité à généraliser sur les données non vues durant l'entraînement. Enfin, pour les différents modèles, l'ensemble de test a été utilisé pour évaluer les performances de chaque classifieur sur un ensemble de données qui n'a pas été utilisé lors de la création du modèle prédictif. Cette figure (voir Figure 2) nous montre que les classes sont bien équilibrées (15120 observations uniformément réparties entre les sept classes), cela signifie qu'il s'agit ici d'un problème de classification multi-classes équilibré.

Nous avons par la suite jugé nécessaire de faire du *feature engineering* en ajoutant une variable correspondant à la distance euclidienne calculée à partir de **Vertical_Distance_To_Hydrology** et de **Horizontal_Distance_To_Hydrology** et une autre variable **Total_Horizontal_Distance** correspondant à la somme de toutes les trois distances horizontales. Celà a eu un impact significatif dans nos résultats.

La matrice de corrélation entre les variables continues est donnée sur la figure 4. Certaines des variables se révèlent être corrélées, notamment les variables associées à la mesure relative de la lumière solaire incidente enregistrée à différents moments, qui montrent des niveaux d'association entre elles.

Pour éviter le surapprentissage des données d'entraînement, nous utilisons la technique de validation croisée k-fold, qui est généralement utilisée pour estimer les performances des résultats qu'une méthode peut atteindre lorsqu'elle est évaluée sur des données indépendantes des données d'entraînement. L'*accuracy* globale a été calculée en prenant la moyenne de chaque type correctement prédit divisé par le nombre total d'observations de tous les ensembles de validation croisée à quatre plis.

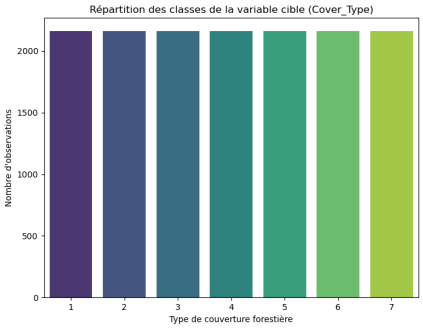

Figure 2 – Répartition des classes

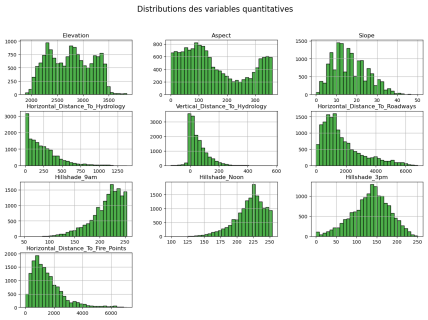

Figure 3 – Distribution des variables

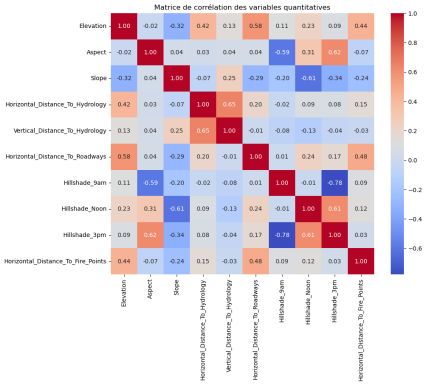

Figure 4 – Matrice de corrélation

2
MODÈLE XGBOOST

Lors de la recherche des modèles les plus adaptés pour effectuer nos prédictions, le modèle XGBoost s'est distingué comme l'un des plus prometteurs dans notre cas. Étant donné la nature détaillée de nos données, les modèles basés sur les arbres de décision étaient particulièrement bien adaptés à nos besoins. Nous avons donc implémenté XGBoost, un algorithme de boosting de gradient qui utilise des arbres de décision comme modèles de base.
Le principe de fonctionnement de ce modèle est d'améliorer les algorithmes traditionnels de gradient boosting. Il construit les modèles de manière séquentielle, en optimisant les erreurs résiduelles du modèle précédent. Les arbres de décision sont construits de manière additive, chaque nouvel arbre corrigeant les erreurs des arbres précédents.

Le modèle de XGBoost tient compte de plusieurs hyperparamètres essentiels :

— **La profondeur des arbres** : contrôle la capacité de chaque arbre à capturer des interactions complexes entre les variables.

— **Le nombre d'estimateurs (arbres)** : détermine le nombre total d'arbres construits.

— **Le taux d'apprentissage (learning rate)** : ajuste la contribution de chaque arbre à la prédiction finale.

Bien que XGBoost fonctionne efficacement avec des valeurs par défaut raisonnables pour ses hyperparamètres, il est souvent nécessaire d'effectuer une recherche approfondie pour trouver la combinaison optimale, en utilisant des techniques comme **GridSearchCV**.

Pour entraîner le modèle, nous avons divisé notre jeu de données en deux ensembles :

— **Ensemble d'entraînement** : utilisé pour construire le modèle.

— **Ensemble de validation** : utilisé pour évaluer les performances du modèle.

Cette division garantit que les performances évaluées reflètent mieux la capacité du modèle à généraliser sur des données inconnues.
Pour la recherche d'hyperparamètres à l'aide de **GridSearchCV**, nous avons utilisé les combinaisons suivantes :

— **Profondeur des arbres** : {10, 15, 20}.

— **Taux d'apprentissage** : {0.01, 0.1}.

— **Nombre d'estimateurs (arbres)** : {100, 200, 300}.

— **Fraction des échantillons (subsample)** : {0.8, 1.0}.

— **Fraction des caractéristiques (colsample_bytree)** : {0.8, 1.0}.

Cependant, cette recherche était coûteuse en temps, car elle impliquait un processus de validation croisée avec un nombre de plis ($CV = 5$). Cela a considérablement alourdi le temps d'exécution et, dans certains cas, provoqué des limitations liées aux ressources de notre machine (plantage du noyau). Pour pallier ces problèmes, nous avons restreint chaque hyperparamètre à trois valeurs possibles et avons utilisé un nombre réduit de plis pour la validation croisée.

Après avoir déterminé les meilleurs hyperparamètres, nous avons entraîné le modèle final sur l'ensemble d'entraînement complet. La validation a montré que XGBoost pouvait offrir une performance solide avec des hyperparamètres bien optimisés.

Après l'entraînement, nous pouvons connaître l'importance des différents *features* dans l'entraînement. Ceux-ci sont présentés dans la figure 5, où nous pouvons remarquer l'importance du fait que nous avons créé de nouvelles variables.

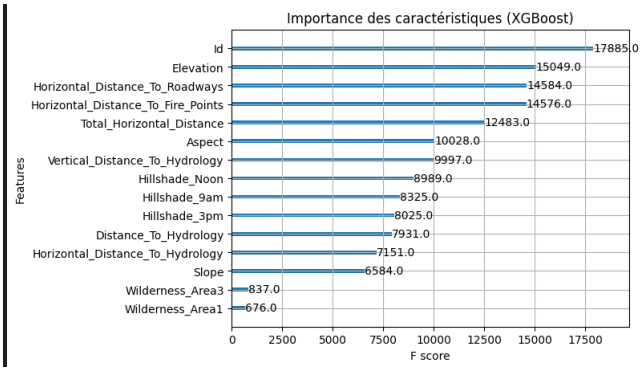

Figure 5 – features importants pour xgboost

3
MODÈLE LIGHT GBM

Toujours en quête d'un modèle performant, nous avons choisi d'explorer **LightGBM** (*Light Gradient Boosting Machine*), un algorithme de boosting spécialement conçu pour traiter de grands volumes de données de manière efficace. Avec un ensemble de données d'entraînement comportant **56 colonnes**, LightGBM s'est révélé être un candidat prometteur grâce à sa capacité à gérer des ensembles de données volumineux tout en offrant des performances rapides.

LightGBM repose sur le principe du **boosting par gradient**, similaire à XGBoost, où des arbres de décision sont construits de manière séquentielle pour corriger les erreurs des prédictions précédentes. Cependant, LightGBM se distingue par des optimisations spécifiques qui le rendent plus efficace :

— **Histogram-based learning** : LightGBM regroupe les valeurs des caractéristiques en bacs (ou histogrammes) pour accélérer les calculs et réduire la mémoire utilisée.

— **Leaf-wise growth** : Contrairement à la croissance de l'arbre par niveau utilisée par XGBoost, LightGBM utilise une croissance par feuille. Cela permet de construire des arbres plus profonds avec moins de nœuds inutiles, ce qui améliore la précision tout en restant rapide.

Comme pour XGBoost, l'entraînement de LightGBM nécessite une recherche d'hyperparamètres pour optimiser ses performances. Ces paramètres incluent notamment :

— La profondeur maximale des arbres.

— Le taux d'apprentissage

— Le nombre d'estimateurs

Tous les paramètres sont pareils que ceux du XGBoost sauf le nombre d'estimateurs dans lequel nous avons remplacé 300 par 1500.

Nous avons rencontré des défis similaires à ceux observés avec XGBoost, tels que des temps d'entraînement prolongés et des blocages du noyau, dus à la recherche d'hyperparamètres intensifs via **GridSearchCV**. Toutefois, LightGBM s'est révélé légèrement meilleur que XGBoost en termes de performance globale. Nous avons également noté que LightGBM utilisait un nombre d'estimateurs significativement plus élevé que XGBoost pour atteindre ses performances optimales. Cela met en évidence sa capacité à exploiter un grand nombre d'arbres tout en restant rapide et efficace.

Comme pour XGBoost, nous pouvons, une fois de plus, observer après l'entraînement l'importance du *feature engineering* effectué lors du prétraitement des données.

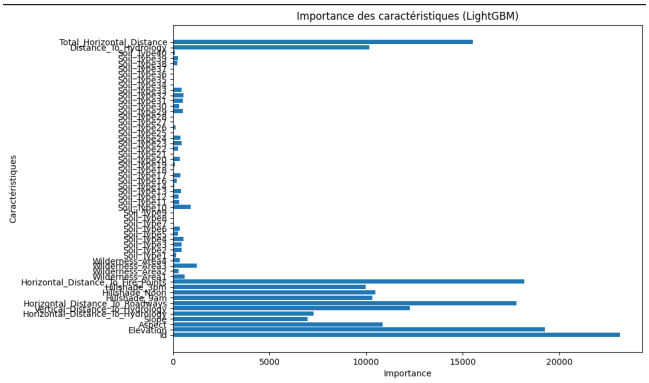

Figure 6 – features importants pour Lightgbm

4 MODÈLE HYBRIDE

Dans notre quête constante d'améliorer les performances de notre modèle et d'obtenir un meilleur score sur les données de test, nous avons envisagé de **combiner les deux modèles précédents** (XGBoost et LightGBM). Cette approche, appelée **stacking**, consiste à utiliser plusieurs modèles de base pour capturer différentes relations dans les données et à les combiner via un modèle de méta-apprentissage.

Nous avons choisi d'utiliser un **Random Forest** comme modèle de méta-apprentissage, en restant fidèles aux modèles basés sur les arbres de décision, qui se sont avérés particulièrement performants dans notre cas. Bien que nous ayons testé d'autres modèles, tels qu'une régression logistique comme méta-modèle, le Random Forest a montré des performances supérieures.

Le stacking repose sur une architecture en deux niveaux :

1. **Niveau 1 : Modèles de base** Les modèles de base (ici **XGBoost** et **LightGBM**) apprennent à partir des données d'entraînement. Chaque modèle génère des prédictions sur les données, et ces prédictions servent ensuite de nouvelles caractéristiques pour le méta-modèle.
2. **Niveau 2 : Méta-modèle** Le méta-modèle (ici un **Random Forest**) apprend à partir des prédictions des modèles de base. Son rôle est de combiner ces prédictions pour produire une estimation finale plus précise.

Pour implémenter cette technique, nous avons suivi les étapes suivantes :

— **Sélection des modèles de base** : Nous avons utilisé les modèles XGBoost et LightGBM, déjà optimisés grâce à une recherche d'hyperparamètres.

— **Choix du méta-modèle** : Nous avons opté pour un Random Forest en raison de ses capacités à gérer les relations non linéaires et sa robustesse. Nous avons utilisé les **paramètres par défaut**, sans effectuer de recherche approfondie d'hyperparamètres, car notre objectif était principalement de combiner les forces des modèles de base.

— **Entraînement du Stacking Model** : Le modèle a été entraîné sur les données de validation croisées générées par les modèles de base, ce qui a permis d'éviter tout surapprentissage.

Le modèle de stacking avec Random Forest a surpassé les performances individuelles de XGBoost et LightGBM, confirmant l'efficacité de cette méthode de combinaison. Les prédictions produites par ce modèle ont mieux capturé les complexités des données grâce à la complémentarité des modèles de base.

Cette amélioration illustre la puissance du stacking, qui exploite la diversité des modèles de base pour produire des prédictions plus robustes et précises. En combinant les modèles XGBoost et LightGBM via un méta-modèle Random Forest, nous avons obtenu une solution qui capitalise sur les forces de chaque algorithme.

5 RÉSULTATS ET COMMENTAIRES

L'accuracy pour le type i est calculée comme suit : n_i / N_i où n_i est le nombre de prédictions correctes pour le type i et N_i est le nombre total de prédictions pour le type i.

L'accuracy globale moyenne de classification et l'accuracy par classe provenant d'une validation croisée à 4 plis pour les trois modèles.

Les précisions de classification produites par chaque modèle, calculées à partir de l'ensemble de test, sont présentées dans cette section.

Modèle Hybride

La figure 7 montre la précision obtenue sur les données de validation et nous permet d'avoir une petite idée sur les différentes classes qui seront mieux prédites, en occurrence les classes Cottonwood/Willow et Krummholz.

	precision	recall	f1-score	support
1	0.83	0.82	0.83	427
2	0.83	0.79	0.81	464
3	0.90	0.93	0.91	400
4	0.96	0.97	0.97	436
5	0.92	0.94	0.93	452
6	0.92	0.91	0.92	415
7	0.96	0.97	0.97	430
accuracy			0.91	3024
macro avg	0.90	0.91	0.91	3024
weighted avg	0.90	0.91	0.90	3024

Figure 7 – Performance du modèle hybride

Cependant, en évaluant notre modèle sur les données de tests (pas encore vues par le modèle) on obtient une *accuracy* globale de classification de 81.94%. En modifiant le méta modèle par la régression logistique, on a une performance de 81.69%. Dans le meilleur des cas, si on ne tient pas compte des colonnes à faibles variances et avec pour meta modèle le *random forests*, on gagne considérablement en performance avec **82.13%**. Tout ceci montre l'intérêt du choix d'un bon meta modèle, des variables pertinentes et même du feature engineering.

Autres modèles

Lorsque nous implémentons les autres modèles comme le *random forests* (RF) ou le *Decision Tree*, on obtient respectivement 79.4% et 71.95% d'*accuracy*. De même, le modèle XGBoost évalué seul fourni une précision de 79%, d'où l'idée d'essayer de corriger les erreurs de prédiction qu'il commet en le combinant avec un autre modèle.

CONCLUSION

Cette étude évalue l'utilité de plusieurs modèles d'apprentissage automatique, associés à différentes méthodes de sélection de variables, pour classifier avec précision les types de couverture forestière. Les modèles XGBoost et Light GBM ont été utilisés pour la classification, accompagnés de diverses techniques de sélection de caractéristiques. Il a été observé que la combinaison de ces deux modèles permet d'atteindre une meilleure précision par rapport aux autres modèles testés, tant pour l'évaluation globale que pour les types de couverture individuels. Cela met en évidence l'efficacité des modèles hybrides dans l'amélioration de la précision des prédictions.

BIBLIOGRAPHIE

- Blackard, J. A., and D. J. Dean. 1999. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. Computers