Rapport Kaggle

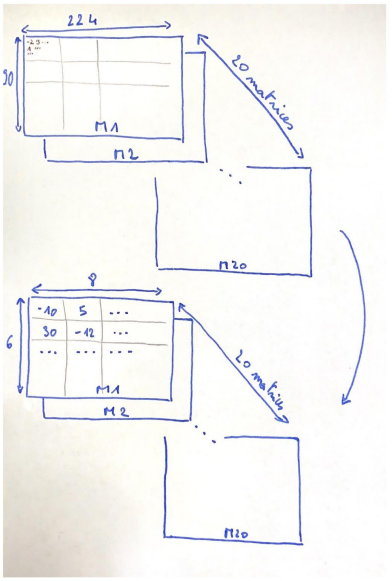
Benjamin Moron et Thibaut Desix

I / Objectif

L'objectif est d'analyser et de classifier des données générées par une caméra d'événements, par tous les moyens possibles. Le jeu de données de lecture labiale utilisé a été collecté au laboratoire i3s pendant l'été 2022 avec une caméra VGA GEN3.1, et il a été nettoyé et prétraité (centré / recadré / redimensionné sur la région de la bouche) pendant l'hiver 2023.

II / Pré-traitement des données

Les données représentent les mouvements de lèvres d'individus prononçant différents mots, ces mouvements sont observés par le changement de polarité au cours du temps extraits des vidéos enregistrant les mouvements de lèvre.
Nous avons reçu des données sous la forme de tableau d'éléments avec en colonnes x et y caractérisant la position d'un pixel de la vidéo, la 3ème colonne représente le changement de polarité : si le pixel est brillant la valeur sera 1 et sinon 0. Le moment où ces changements se produisent est quantifié dans une 4ème colonne de temps noté t.
La difficulté résidant dans ce prétraitement est de pouvoir garder des données significatives tout en les rendant plus synthétiques et interprétables par des méthodes de classification par la suite. Nous avons décidé tout d'abord de transformer les données en matrices afin de récupérer l'information spatiale des données pour la synthétiser plus facilement par la suite.
La première étape a été de stocker dans un tenseur les données avec en 1ère coordonnée le temps et en 2ème et 3ème les coordonnées x et y. Ceci permet d'observer à un temps t des matrices représentant comme une capture d'écran de l'enregistrement stockant les changements de polarité.
Ce format de données a été par la suite synthétisé de la manière suivante :
Des subdivisions de temps et d'espace ont été créées (20 en temps, 28 en x et 16 en y) afin de résumer les changements généraux de polarité sur la zone pendant cette période.
Nous avons rassemblé les informations temporelles des matrices en ajoutant -1 à chaque apparition de 0 et +1 à chaque 1 dans la polarité. Ceci nous a donné un tenseur de longueur 20 de matrices de taille 224 x 90. Nous avons répété cette opération au sein de "blocs" de subdivisions spatiales réformés dans une matrice de taille (224/28 x 90/15). Le choix du nombre de subdivisions a été choisi pour être un diviseur des dimensions respectives de la résolution en x et y.
Le passage en bloc est illustré par le schéma ci-contre :

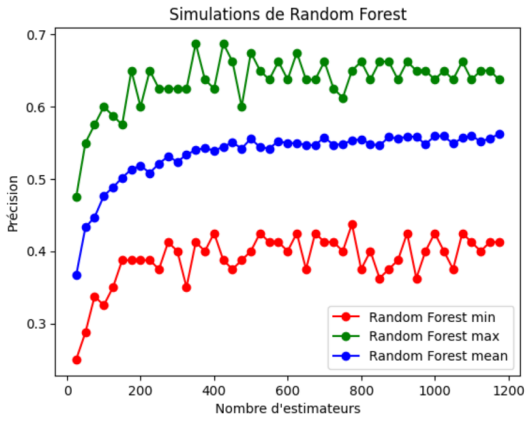


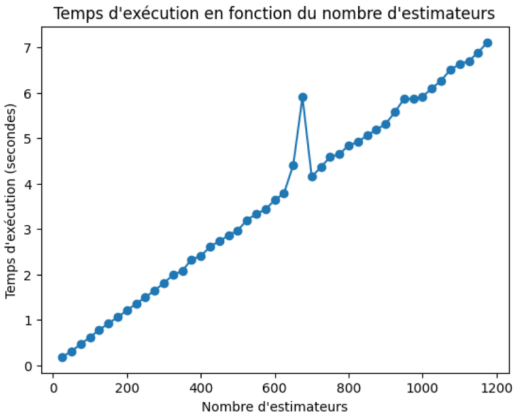
III / Méthode de classification

1. Random Forest

La Forêt Aléatoire est un algorithme d'apprentissage ensembliste qui combine plusieurs arbres de décision, construits de manière aléatoire, pour améliorer la précision et la robustesse des prédictions. Ses avantages incluent la réduction du surajustement, la gestion efficace de données complexes, et la flexibilité dans la modélisation.

Nous avons donc dans un premier temps utilisé une classification par Random Forest et observé les précisions obtenues en fonction du nombre d'estimateurs qu'utilise la méthode.



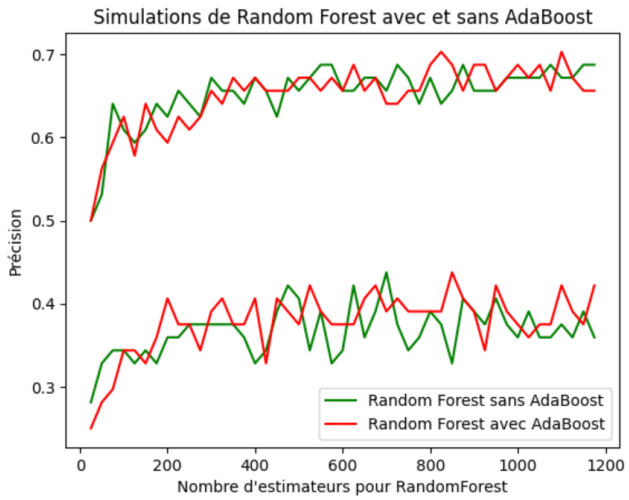


Dans le graphique illustrant les précisions maximales, minimales et moyennes résultant de 30 simulations de Random Forest pour chaque nombre d'estimateurs, une observation se dégage : à mesure que le nombre d'estimateurs augmente, la précision tend à croître. Toutefois, cette amélioration de la précision s'accompagne d'une augmentation du temps d'exécution. De manière notable, au-delà d'un seuil spécifique, la précision semble atteindre un plateau, stagnant autour de 0.65 pour les meilleures simulations. Ces observations soulignent un compromis entre la précision du modèle et le coût en termes de temps d'exécution, mettant en évidence l'importance de trouver un équilibre optimal dans le choix du nombre d'estimateurs pour un modèle Random Forest.

2. Random Forest avec AdaBoost

AdaBoost, ou Adaptive Boosting, est un algorithme d'apprentissage ensembliste qui vise à améliorer la performance des modèles de faible qualité en leur accordant des poids différents. Pendant l'entraînement, il ajuste itérativement les poids des observations mal classées, donnant ainsi plus d'importance aux erreurs. Cela permet à AdaBoost de se concentrer sur les observations difficiles et de construire un modèle fort à partir de plusieurs modèles faibles. Les avantages d'AdaBoost incluent une meilleure capacité à gérer des données bruitées, une réduction du surajustement, et une amélioration de la précision globale du modèle.

Nous avons ensuite ajouté une méthode AdaBoost au RandomForest pour en comparer l'efficacité.



Ce graphique illustre les précisions maximales minimales résultant de 30 simulations de Random Forest avec et sans AdaBoost pour chaque nombre d'estimateurs que prend Random Forest. Nous avons fixé le nombre d'estimateurs que prend AdaBoost fixé à 30 puisque comme pour Random Forest, l'efficacité du modèle atteint un seuil et l'augmentation de ce paramètre augmentera le temps d'exécution sans avoir d'impact sur le résultat.
Nous remarquons alors que la classification par Random Forest sans AdaBoost est aussi efficace que celle avec AdaBoost. Cependant, l'utilisation d'AdaBoost augmente le temps d'exécution donc la méthode que nous avons retenue des deux et celle ne l'utilisant pas.

3. Extra Trees

La méthode Extra Trees est une variante de la forêt aléatoire. Elle construit un ensemble d'arbres de décision en introduisant davantage d'aléatoire dans le choix des seuils de séparation et des caractéristiques utilisées, visant ainsi à renforcer la diversité des arbres et à améliorer la capacité de généralisation du modèle.
Cette méthode a été testée lors d'essais de voting, et elle donnait des résultats d'accuracy comparables au random forest.

4. Voting

La méthode de Voting en apprentissage machine est une technique d'ensemble où plusieurs modèles sont combinés pour prendre des décisions de manière collective. Elle peut être réalisée en utilisant des stratégies telles que le vote majoritaire (hard voting) ou la moyenne des probabilités (soft voting), fournissant ainsi une prédiction finale basée sur la contribution de chaque modèle, pondérée ou non. Dans notre cas, la différence entre les accuracy obtenues avec hard et soft voting étaient comparables, nous avons arbitrairement choisi le "hard voting". Cette méthode n'a pas beaucoup amélioré les résultats obtenus avec le random forest car cette méthode et Extra tree donnaient une accuracy nettement supérieure aux autres méthodes testées. Nous avons essayé dans un premier temps de mettre une pondération proportionnelle à l'accuracy sans grand changement, cela correspondait à la tentative numéro 3. Ce procédé nous a permis en plus de visualiser l'accuracy de chaque "voter" et d'éliminer ceux jugés trop peu précis (< 50%). La conclusion est que par manque de features les méthodes de classifications donnaient des prédictions basées sur les mêmes types d'interprétation de la donnée, et multiplier les classifications revenaient à prédire plusieurs fois la même chose et de voter pour ce choix commun.

5. Gradient boosting et Extrem Gradient Boosting

Le Gradient Boosting est une méthode d'apprentissage ensembliste qui combine plusieurs modèles faibles, généralement des arbres de décision, pour créer un modèle robuste. Il ajuste les modèles de manière séquentielle en se concentrant sur les erreurs résiduelles, améliorant ainsi la précision globale. Ses avantages comprennent une grande précision, une capacité à traiter des données complexes, et une flexibilité accrue.

L'Extreme Gradient Boosting (XGBoost) est une amélioration du Gradient Boosting, reconnue pour sa rapidité, sa performance, et sa capacité à gérer efficacement les données manquantes. Ses avantages comprennent une grande précision et une adaptabilité à divers problèmes d'apprentissage automatique.

L'utilisation du Gradient Boosting et de l'Extreme Gradient Boosting nous donnait des résultats trop faibles aux alentours de 0.40 de précison, c'est pourquoi nous avons choisi de ne pas les utiliser. Cela pourrait s'expliquer sur le type de données que nous utilisons qui ne serait pas adapté pour ces méthodes de boosting.

6. Recherche des meilleurs paramètres de classification

La recherche des meilleurs paramètres de classification, effectuée à l'aide de la méthode GridSearch, consiste à explorer de manière systématique différentes combinaisons de paramètres pour chaque modèle. GridSearch évalue la performance du modèle sur un ensemble prédéfini de paramètres, permettant ainsi d'identifier la configuration optimale qui maximise la précision du modèle.
Nous avons donc cherché à estimer quels pourraient être les meilleurs paramètres pour la méthode Random Forest. Nous avons trouvé que la précision maximale était atteinte en premier pour un nombre d'estimateur à 600.

IV / Conclusion

La méthode que nous avons trouvée la plus efficace par ses résultats et par son temps d'exécution est la méthode de classification utilisant Random Forest sans boosting.
Notre meilleur précision obtenue en local a été de 0.65, et plusieurs étapes de notre travail auraient pu être améliorées afin d'affiner la classification de ces données.
Tout d'abord le prétraitement, une chose à laquelle nous avons pensé trop tard est d'affecter des poids à des régions plus sollicitées que d'autres. Deuxièmement, l'interprétation des données semblait plutôt adaptée à une classification par réseaux de neurones convolutionnels tels que le CNN ou le RNN, mais le format de données à notre disposition suite au pré-processing opéré rendait la classification inefficace.