RER – Random Forest

# Contexte :

Découvrir le modèle de machine learning et ses concepts de base des random forest.

# Problématique(s) :

* Comment créer et évaluer un modèle random forest
* Comment optimiser le modèle
* Qu'est-ce que la régularisation ? Quelles sont les techniques de régularisation ?

# Mots clés :

* Méthode d’ensemble : combiner les prédictions de plusieurs estimateurs de base construits avec un algorithme d'apprentissage donné afin d'améliorer la généralisabilité/robustesse sur un seul estimateur.
* RandomForest Classifier : Un classificateur de forêt aléatoire est un type de méthode d'apprentissage d'ensemble pour la classification (et la régression) qui construit plusieurs arbres de décision au moment de la formation et produit la classe qui est le mode des classes (classification) ou la prédiction moyenne (régression) des arbres individuels. L'idée derrière la forêt aléatoire est de décorréler les arbres en entraînant chacun sur un sous-ensemble aléatoire de données, puis de faire la moyenne des résultats. Cela réduit le surajustement et améliore la précision du modèle.
* feature\_importances : feature\_importances est une technique utilisée pour déterminer l'importance relative de chaque fonctionnalité dans un jeu de données, qui peut être utilisée pour sélectionner un sous-ensemble des fonctionnalités les plus importantes à utiliser dans un modèle, ou pour comprendre quelles fonctionnalités déterminent les prédictions d'un modèle.
* Bagging : Bagging signifie Bootstrap Aggregating. Il s'agit d'une technique utilisée pour améliorer la stabilité et la précision des modèles d'apprentissage automatique, en particulier les arbres de décision. Le bagging fonctionne en formant plusieurs copies du même modèle, chacune sur un sous-ensemble aléatoire différent des données de formation. Ces sous-ensembles sont créés en échantillonnant de manière aléatoire les données d'apprentissage avec remplacement. La prédiction finale est faite en faisant la moyenne des prédictions de tous les modèles individuels. Cela réduit la variance du modèle, le rend moins sujet au surajustement et améliore ses performances de généralisation. Le bagging est un moyen de fournir un caractère aléatoire dans le modèle en formant plusieurs versions de celui-ci.
* KneighborsClassifier : Il s'agit d'un type d'algorithme d'apprentissage automatique supervisé utilisé pour la classification. L'idée de base derrière le classificateur k-NN est qu'il classe un nouveau point de données en fonction de sa proximité avec d'autres points de données dans l'ensemble d'apprentissage. Pour ce faire, il trouve le nombre k de voisins les plus proches du nouveau point de données, puis classe le nouveau point de données en fonction de la classe majoritaire de ses k voisins les plus proches. La valeur de k est généralement un petit entier positif, tel que 5 ou 10, mais peut être n'importe quel entier positif. La métrique de distance utilisée pour déterminer la proximité des points de données peut être la distance euclidienne, la distance de Manhattan ou toute autre métrique de distance.
* Pasting : Le Pasting est une variante debagging utilisée pour l'apprentissage d'ensemble. La principale différence entre le bagging et le pasting est que dans le bagging, les données de formation sont échantillonnées avec remplacement, alors que dans le collage, les données de formation sont échantillonnées sans remplacement.
* Out-Of-Bag : Out-of-Bag est une méthode utilisée pour estimer l'erreur de généralisation d'un modèle d'ensemble, en évaluant la précision du modèle sur les instances qui n'ont pas été utilisées dans la formation de chaque modèle.
* Random Subspaces : Random Subspace est une variante de la technique d'apprentissage d'ensemble qui combine le bagging et la sélection de caractéristiques. L'idée principale derrière les sous-espaces aléatoires est de sélectionner un sous-ensemble aléatoire de caractéristiques et d'entraîner plusieurs modèles, chacun sur un sous-ensemble aléatoire différent des données d'entraînement, puis d'agréger les prédictions des modèles individuels.
* Random Patches : Random Patches est une méthode d'apprentissage d'ensemble qui combine le bagging et l'extraction de caractéristiques. Il est similaire aux sous-espaces aléatoires, mais au lieu de sélectionner un sous-ensemble aléatoire d'entités, il sélectionne des patchs aléatoires de l'espace d'entités. Dans Random Patches, plusieurs modèles sont entraînés sur différents patchs aléatoires de l'espace des fonctionnalités. Un patch est défini comme un sous-ensemble de l'espace des fonctionnalités qui comprend un certain nombre de fonctionnalités et leurs interactions. La prédiction finale est faite en faisant la moyenne des prédictions de tous les modèles individuels.
* Optimisation
* Régularisation
* Gridsearch :
* GridsearchCV : GridsearchCV est une méthode utilisée pour ajuster les hyperparamètres d'un modèle en effectuant une recherche exhaustive sur une grille de paramètres spécifiée, en évaluant le modèle pour chaque combinaison de paramètres et en renvoyant l'ensemble de paramètres qui donne les meilleures performances.
* Bootstrap

# Hypothèses :

* Osman random forest est plus rapide et plus précis que ID3
* Jean-Paul dans random forest il n’y a pas de racine, c’est une forêt sans racine.
* Tetyana Random Forest peut être utilisé avec toutes sortes de données et dans tous les domaines de classification
* Aude les racines des random forest sont choisi aléatoirement
* Etienne gridsearch permet de scanner l’hyper espace en machin learning
* Adeline les modèles randoms forest ne sont pas tous performant selon les domaines d’activités (données)
* Briand random forest ne donne pas de sortie précise mais une approximation en fonction de la moyenne majoritaire des arbres décisionnels
* Seydou les arbres de RF n’ont pas tous le même poids
* Axel
* Loïc : Random forest n'est ni plus ni moins qu'un assemblage de plusieurs Decision Tree.
* Nicolas le bootstrap du bagging est le même bootstrap d’internet
* Solenn le features importance dans scikitlearn permet de donner un poid dans un arbre d’une forêt
* Adrien les random forest les arbres sont choisis aléatoirement
* Adeline la prédiction finale d’un modèle random forest est somme (ou moyenne) de décision aléatoire.
* Loïc On utilise le random forest pour augmenter le nombre de génération de decision tree afin d'en sortir un résultat plus précis encore que le decision tree
* Briand le bagging est pasting constitent à regrouper des arbres par correspondance
* JP II il faut un minimum d’attribut pour générer une Random Forest

# Plan d'action :

* Explorer les ressources
* Définir et comprendre les mots clefs
* Faire le Workshop
* Tableau comparatif entre random forest et arbre de décision
* Utilisation du gridsearch et validation croisée pour l'optimisation
* RER