Projet A3

Intelligence Artificielle

Mathys Bodennec – Nicolas Guermeur

Table of Contents

Contexte du projet

Ce projet est la seconde partie du grand projet Big Data/IA/Web. Le but de ce grand projet est d'appliquer les compétences acquises dans divers modules (Big Data, Intelligence Artificielle, Développement Web) à une étude complète sur le patrimoine arboré de la ville de Saint-Quentin (Aisne). L'objectif principal est de concevoir et développer une application permettant de traiter et de visualiser les données des arbres, et d'identifier ceux nécessitant une attention particulière, comme les arbres à abattre.

Objectifs IA

- Comprendre les étapes d'un projet d'apprentissage automatique
- Apprendre à évaluer une méthode d'apprentissage automatique
- Manipuler les différentes techniques d'apprentissage automatique:
  - Apprentissage non supervisé: clustering
  - Apprentissage supervisé: classification, régression

Cahier des charges

1. Préparation des données
2. Apprentissage non supervisé
3. Apprentissage supervisé
4. Métriques
5. Création de scripts utilisables en ligne de commande

Réalisation

Préparation

Lors de ce projet, nous avons utilisé VSCode pour le développement, et GitLab pour la gestion de version. La répartition des tâches (sous forme de diagramme de Gantt) est disponible en Annexe 1.

Besoin client 1: Visualisation sur carte

- Créer un visuel sur une carte qui sépare les arbres selon leur taille.
- L'utilisateur pourra choisir le nombre de catégories (par exemple, deux catégories: petits et grands, ou trois catégories: petits, moyens et grands).

Préparation des données

- Extraction des données d'intérêt: Séléctionner les colonnes de la base de données selon ce besoin.
- Encodage des données catégorielles si nécessaire: Utiliser des techniques de prétraitement pour convertir les données non numériques en données numériques si nécessaire.

Les colonnes sélectionnées sont: longitude, latitude, tronc_diam, haut_tronc. La longitude, et la latitude sont pour placer les arbres sur la carte; les 2 autres variables sont pour entrainer les modèles pour estimer la taille de l'arbre. Justifier

Étant donné que toutes ces variables sont numériques, il n'y a pas besoin de le prétraiter.

Apprentissage non supervisé

- Choix de l'algorithme de clustering: Sélectionner un/des algorithme(s) de clustering pour séparer les arbres en groupes basés sur leur taille.
- Détermination du nombre de clusters: Expérimenter avec différents nombres de clusters pour voir quel modèle offre la meilleure séparation des tailles d'arbres.

Les algorithmes étudiés: KMeans, Regroupement Hiérarchique Ascendant (Agglomerative Clustering), Partitionnement Spectral (Spectral Clustering).
Le **K-means** est un algorithme qui vise à partitionner n observations en k clusters de manière que chaque observation appartienne au cluster dont le centre est le plus proche. L'approche consiste à minimiser la somme des distances entre les observations et les centres des clusters.
L'**Agglomerative Clustering**, est un type de clustering hiérarchique; qui est, comme son nom l'indique, un algorithme qui cherche à former une hiérarchie entre les clusters. Dans le cas qui nous intéresse, au début, chaque observation commence dans son propre cluster; puis l'algorithme fusionne des paires de clusters les plus proches (ici mesuré avec la distance de Ward) au fur et à mesure qu'on monte dans la hiérarchie.
Le **Spectral Clustering** est un algorithme qui utilise les valeurs de la matrice de corrélation pour réduire la dimensionnalité des données et former les clusters avec ces dimensions réduites.

Métriques pour l'apprentissage non supervisé

- Évaluation des clusters: Utiliser des métriques pour évaluer la qualité du clustering.

Pour mesurer l'efficacité des algorithmes, on calcule 3 valeurs: le silhouette score, le Calinski-Harabasz score, et le Davies-Bouldin score. On veut que les 2 premières valeurs soient élevées, et que la dernière soit faible.
Graphs
On choisit l'Agglomerative Clustering.

Visualisation sur carte

- Création de la carte: Utiliser une bibliothèque de visualisation pour représenter les arbres sur une carte avec des couleurs différentes pour chaque cluster.

La bibliothèque utilisé pour la visualisation est plotly.
Carte

Préparation de script

- Développement de la fonction principale: Écrire une fonction qui prend en entrée le choix de l'utilisateur (nombre de catégories) et qui retourne une carte avec les positions des arbres colorées selon le résultat du clustering.

Attendus
Justifier le choix des variables sélectionnées
Justifier le choix du modèle de clustering + Expliquer le principe de son fonctionnement
Justifier le choix des métriques et discuter les résultats obtenus (Tableau + graphes)

Besoin client 2: Modèle de prédiction de l'âge

Préparation des données

- Extraction des données d'intérêt: Sélectionner les colonnes pertinentes de la base de données selon ce besoin.
- Encodage des données catégorielles.
- Normalisation des données ou pas.
- Cible: `('age_estim')`

Les colonnes sélectionnées sont: haut_tronc, tronc_diam, fk_prec_estim, age_estim. Toutes ces variables ont une corrélation avec l'âge supérieur à 0.6.
On normalise age_estim pour avoir de meilleurs résultats.

Apprentissage supervisé pour la régression

- Choix de l'algorithme d'apprentissage de la classification: Utiliser des métriques pour évaluer la qualité du modèle de la classification.

Plusieurs algorithmes différents ont été testés et comparés: Forêt d'Arbres Décisionnels (Random Forest), K plus proches voisins (KNN Regression), Bagging, et Boosting.
Random Forest, est une technique d'apprentissage qui construit initialement plusieurs arbres de décision. Pour les tâches de régression, la prédiction moyenne des arbres individuels est utilisée.
KNN Regression, est un algorithme d'apprentissage supervisé sans paramètres. Dans le cas de la régression la prédiction est la moyenne des valeurs des K plus proches voisins. Si les valeurs qu'il doit prédire sont très différentes entre elles, normaliser le résultat améliore la précision drastiquement.
Bagging, ou Bootstrap aggregating, est un méta-algorithme qui a pour but d'améliorer la stabilité et la précision des algorithmes d'apprentissage machine. Ce méta-algorithme est souvent basé sur des arbres de décision et y ajoute de l'aléatoire dans sa construction. Il permet de réduire l'overfitting.
Boosting, une technique d'apprentissage automatique qui fait partie des méthodes d'ensemble. Il s'agit d'un algorithme qui combine plusieurs modèles faibles (comme des arbres de décision peu profonds) pour créer un modèle fort et performant. L'objectif est d'améliorer progressivement la performance globale en corrigeant les erreurs des modèles précédents.

Métriques pour la régression

- Évaluation de résultats de la classification: Utiliser des métriques pour évaluer la qualité du modèle de la classification.
- Utilisation du GridSearchCV pour tester toutes les combinaisons possibles avec **plusieurs hyperparamètres du modèle** et trouver le meilleur modèle de prédiction.

Pour mesurer l'efficacité des algorithmes, on calcule 3 valeurs: le RMSE (Root Mean Squared Error), le MAE (Mean Absolute Error), et R^2. On veut que les 2 premières valeurs soient faibles, et que la dernière soit proche de 1.
Graphs

Préparation de script

- Développement de la fonction principale: Écrire une fonction qui prend en entrée un **DataFrame de test (.JSON)** et qui retourne la prédiction de l'âge en format .JSON.

Attendus
Justifier le choix des variables sélectionnées (Tableau + graphes)
Justifier le choix du modèle de classification + Expliquer le principe de son fonctionnement
Justifier le choix des métriques et discuter les résultats obtenus (Tableau + graphes)

Besoin client 3: Système d'alerte pour les tempêtes

Préparation des données

- Extraction des données d'intérêt: Sélectionner les colonnes pertinentes de la base de données selon ce besoin.
- Encodage des données catégorielles.
- Normalisation des données ou pas.
- Cible: `('fk_arb_etat')`

Apprentissage supervisé pour la classification

- Choix de l'algorithme d'apprentissage de la classification: Utiliser des métriques pour évaluer la qualité du modèle de la classification.

Métriques pour la classification

- Évaluation de résultats de la classification: Utiliser des métriques pour évaluer la qualité du modèle de la classification.
- Utilisation du GridSearchCV pour tester toutes les combinaisons possibles avec **plusieurs hyperparamètres du modèle** et trouver le meilleur modèle de prédiction.

Préparation de script

- Développement de la fonction principale: Écrire une fonction qui prend en entrée un **DataFrame de test (.JSON)** et qui retourne la prédiction de l'âge en format .JSON.

Attendus
Justifier le choix des variables sélectionnées (Tableau + graphes)
Justifier le choix du modèle de classification + Expliquer le principe de son fonctionnement
Justifier le choix des métriques et discuter les résultats obtenus (Tableau + graphes)
Bonus
Un système qui donne la classification + la probabilité est recommandé