

## ING2 : PROJET INITIATION RECHERCHE

---

# Classification de parcelles agricoles à partir de séries temporelles d'images satellites

---

*Equipe :*

LISA ARGENTO  
CLAIRE GIRARDIN  
ALEXYS REN

*Encadrant :*

ALEXANDRE  
HIPPERT-FERRER

## **Remerciements**

Nous souhaitons adresser nos remerciements à toutes les personnes qui ont apporté leur aide et qui ont contribué à la réalisation de notre projet.

Nous remercions tout d'abord Monsieur Alexandre Hippert-Ferrer pour ses conseils, son soutien et son aide judicieuse tout au long de ce projet.

Nous souhaitons également remercier Marc Rußwurm pour sa disponibilité et les précisions qu'il a su apporter et qui nous ont permis d'avancer sur l'étude technique des données utilisées.

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Données</b>	<b>4</b>
2.1	Présentation des données Breizhcrops . . . . .	4
2.2	Données construites sur le modèle Breizhcrops . . . .	5
<b>3</b>	<b>Méthodes employées</b>	<b>5</b>
3.1	Présentation des algorithmes utilisés . . . . .	5
3.2	Traitement des données et calculs . . . . .	7
<b>4</b>	<b>Résultats et discussion</b>	<b>8</b>
4.1	Étude de la classification . . . . .	8
4.2	Extension du jeu de données . . . . .	10
<b>5</b>	<b>Conclusion et perspectives</b>	<b>10</b>
<b>6</b>	<b>Annexes</b>	<b>11</b>

# 1 Introduction

Les programmes modernes d'observation de la Terre produisent actuellement d'énormes volumes de séries temporelles d'images satellites. Ces images apportent de nombreuses informations, mais extraire des connaissances à partir de ces données constitue encore un objectif majeur de la recherche en traitement des images issues de la télédétection.

Notre étude s'appuie largement sur l'article et les recherches de M. Rußwurm et ses collègues [7]. Dans ces travaux, les auteurs présentent un nouveau jeu de données de parcelles agricoles pertinentes pour mettre en évidence toutes les difficultés que l'on peut rencontrer en faisant de la classification, c'est-à-dire en analysant des données de façon à regrouper les pixels selon leur appartenance à des régions homogènes. Il existe deux principaux types de classification en télédétection : la *classification non supervisée* et la *classification supervisée*. La première permet de regrouper des objets spatiaux en un certain nombre de classes d'occupation du territoire en formant des groupes les plus homogènes possibles tandis que la seconde permet de répartir au mieux ces objets spatiaux dans des classes prédéfinies<sup>1</sup>.

Les auteurs présentent un nouveau jeu de données pour la classification de parcelles agricoles en Bretagne. L'objectif de leur étude est de fournir à la communauté scientifique en *machine learning* un standard pouvant servir de *benchmark* pour de nouvelles méthodes de classification supervisée. L'intérêt est que *Breizhcrops* présente de nombreux défis : données manquantes, déséquilibre de classes, autocorrélation spatiale, etc. Effectivement, pour regrouper des observations en groupes homogènes, il faut tout d'abord avoir une définition de ce que sont des observations similaires ou des observations différentes. Il faut donc être en mesure de quantifier la similarité ou la distance entre deux observations en fonction de leurs attributs. Cette première étape est souvent la plus difficile de tout le processus de classification et c'est souvent sur la définition des groupes que nous jouons pour obtenir la classification la plus précise possible.

Parmi les difficultés que présente le jeu de données *Breizhcrops*, nous nous sommes intéressés à la *sous-représentation*, appelée aussi *déséquilibre* de classes. En effet, un déséquilibre important du nombre de représentants par classe entraîne des prédictions moins robustes car cela augmente nettement la difficulté de l'apprentissage par l'algorithme. Si l'algorithme n'a que peu d'exemples de la classe minoritaire sur lesquels apprendre lors de la phase d'entraînement, il aura des difficultés à prédire les nouveaux objets de cette classe lors de la phase de généralisation (ou *phase de test*). Afin de réduire le déséquilibre de classes, une solution peut consister à multiplier les sources d'observations des échantillons sous-représentés.

Dans le cas de la classification d'occupation des sols agricoles, il pourrait être intéressant d'étudier plusieurs années car la rotation des cultures est fréquente pour ne pas épuiser le sol. L'information temporelle redondante, pluriannuelle, s'avère donc utile car elle permet d'obtenir des classes peu présentes sur une année. Dans le travail de troisième année d'un de nos prédécesseurs [6], l'auteur met en évidence plusieurs méthodes pour exploiter la multi-temporalité des données et ainsi augmenter le nombre d'observations pour des groupes sous-représentés.

---

1. Principe général de la classification en Annexe.

À partir de cet axe de réflexion, notre étude s’articule autour de trois objectifs :

- i) Appliquer certaines méthodes de l’état de l’art pour une tâche de classification supervisée des parcelles agricoles ;
- ii) Contribuer au jeu de données *Breizhcrops* en l’étendant aux années de 2018 à 2022, ce qui n’est pas le cas pour l’instant ;
- iii) Etudier comment la prise en compte des données pluriannuelles permet d’améliorer la classification de parcelles agricoles sous-représentées à partir de cette extension.

L’enjeu est donc de trouver **comment utiliser au mieux la redondance spatiale des cultures créée par l’observation pluriannuelle**. Notre approche est de considérer un même champ différent si les images ne sont pas de la même année.

## 2 Données

### 2.1 Présentation des données Breizhcrops

Un groupe de chercheurs a construit BreizhCrops, un nouveau jeu de données de référence pour la classification supervisée de parcelles agricoles à partir de séries temporelles satellitaires. Ils ont sélectionné neuf types de cultures plus ou moins présents en Bretagne (blé, maïs, orge, colza, tournesol, prairies permanentes, prairies temporaires, vergers, fruits-à-coque) pour mettre en évidence et comprendre les différents problèmes de la classification et entraîner des algorithmes. Ils ont ainsi comparé sept réseaux de neurones profonds sur ce jeu. Ils ont mis en ligne l’ensemble des données ainsi que les (ré)implémentations de modèles et les pondérations de modèles pré-entraînés sur GitHub<sup>2</sup>. Leur objectif était de rendre accessible leur travail, de maintenir le référentiel avec des données supplémentaires et de pouvoir accueillir les contributions de nouvelles méthodes pour construire un jeu de données de référence encore plus pertinent pour la classification et la cartographie des types de cultures.



FIGURE 1 – Départements bretons utilisés pour partitionner les données

La zone étudiée, la Bretagne, située dans le nord-ouest de la France couvre 27 200 km<sup>2</sup>. Les données sont réparties en 5 jeux de données, un par département et Belle-Île-en-Mer (différenciée du Morbihan) comme le montre la fig. 1. Ils ont récupéré les informations des parcelles agricoles bretonnes sur le site de l’IGN : un shapefile contenant toutes les parcelles bretonnes avec l’identifiant de chaque parcelle ainsi que son label, c’est-à-dire la culture majoritaire du champs. Puis, grâce à un code python (`query_gee.py`), ils font appel à la librairie Earth Engine pour ajouter la réflectance, récupérée grâce aux séries temporelles Sentinel-2 sur un an, à chaque champs. L’ensemble des données comprend environ 610 000 observations. Nous avons choisi d’utiliser uniquement les données L1C (Top of Atmosphere) qui ne sont pas corrigées du bruit atmosphérique car la correction n’est pas possible pour certaines régions. Cela permet d’avoir globalement plus de données dans les régions.

2. <https://github.com/dl4sits/breizhcrops>

## 2.2 Données construites sur le modèle Breizhcrops

Sur le même modèle, nous avons récupéré les données des parcelles des années 2018 à 2021, ces parcelles sont celles recensées dans le cadre de la Politique Agricole Commune (PAC). Pour cela il faut télécharger les données RPG de l'IGN<sup>3</sup> pour la région Bretagne pour chaque année. Nous avons fait une sélection par attribut pour garder les mêmes types de culture que les données Breizhcrops. Puis nous avons partitionné les données en réalisant une sélection des parcelles par département. Cela permet de manipuler des données moins lourdes. Enfin, nous avons essayé d'appliquer le même code python pour ajouter la réflectance aux parcelles mais nous avons rencontré plusieurs problèmes quant à l'environnement à utiliser<sup>4</sup> et au chargement très long des données qui nous a fait revoir nos objectifs.

## 3 Méthodes employées

Pour traiter toutes ces données, il nous a fallu choisir des algorithmes adaptés et accessibles. Notre commanditaire nous a conseillé d'utiliser Scikit-Learn [5] : une bibliothèque libre Python destinée à l'apprentissage automatique. Nous avons donc pu découvrir et nous approprier ce nouvel outil.

### 3.1 Présentation des algorithmes utilisés

Pour commencer, nous avons choisi des techniques de Machine Learning très populaires : les  $k$ -nearest neighbors (KNN) [4] et le Random Forest. Le KNN est un algorithme d'apprentissage supervisé qui utilise la proximité pour effectuer des classifications ou des prédictions : il identifie les plus proches voisins d'un point donné pour lui attribuer la classe majoritaire de ces voisins. Il est généralement utilisé comme algorithme de classification, en partant de l'hypothèse que des points similaires peuvent être trouvés les uns à côté des autres : ce qui est le cas lorsque l'on étudie des parcelles agricoles.

Le KNN possède plusieurs paramètres que l'on peut faire varier :

- Le nombre entier de voisins à prendre en compte.
- La métrique utilisée (distance euclidienne, manathan, chebyshev ...) pour identifier les plus proches voisins.
- Le poids choisi pour chaque voisin (poids 'uniformes' i.e. tous les points de chaque voisinage sont pondérés de manière égale ou poids 'distance' i.e. points pondérés par l'inverse de leur distance, dans ce cas les voisins les plus proches d'un point de requête auront une plus grande influence que les voisins qui sont plus éloignés).

Le Random Forest[3] est rapide à entraîner et produit des résultats généralisables. C'est un algorithme qui se base sur l'assemblage d'arbres décisionnels indépendants. Chaque arbre dispose d'une vision parcellaire du problème du fait d'un double tirage aléatoire : un tirage aléatoire avec remplacement sur les observations (*tree bagging*) et un tirage aléatoire sur les variables (*feature sampling*).

---

3. <https://geoservices.ign.fr/rpg>

4. Connexion des comptes Google Earth Engine (GEE)[1] et Google cloud (plateformes de cloud computing)

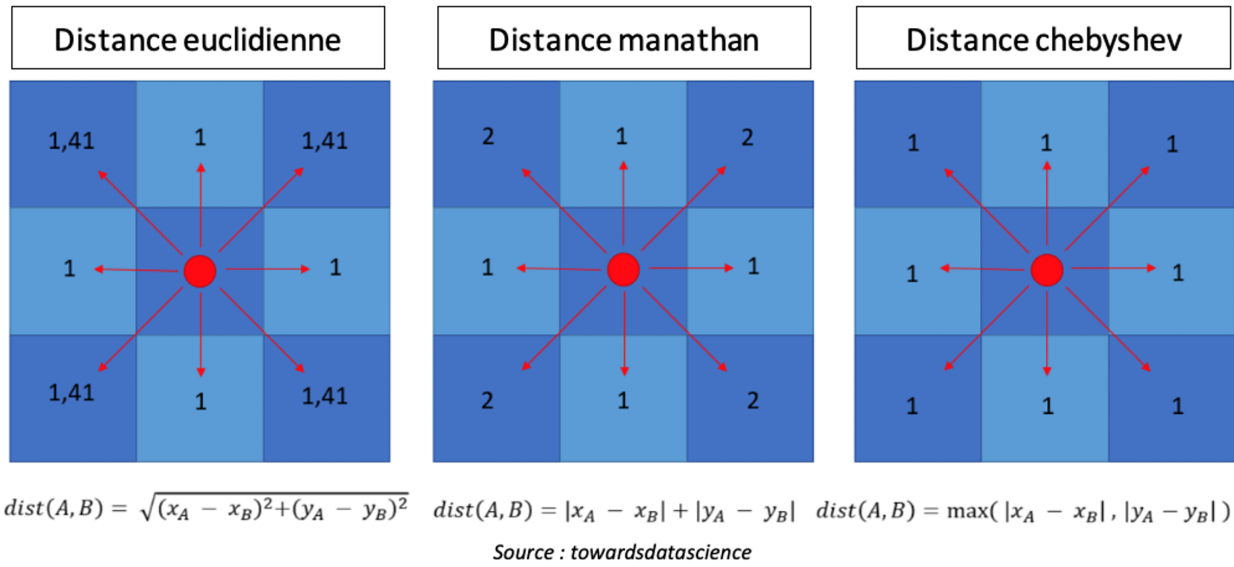


FIGURE 2 – Les différentes métriques utilisables pour le KNN

Nous voulions également des algorithmes moins classiques, notamment utiliser le Deep Learning avec un réseau de neurones récurrents comme le RNN (Recurrent Neural Network). Ce type de réseau de neurones présente des connexions récurrentes : il est constitué de neurones interconnectés possédant des poids et interagissant non-linéairement (avec au moins un cycle dans la structure). Ainsi la sortie d'un neurone est une combinaison non linéaire de ses entrées. Les réseaux de neurones récurrents sont particulièrement adaptés pour l'analyse de séries temporelles. Malheureusement, cet algorithme n'est pas disponible sur scikit-learn. Nous l'avons trouvé sur Pytorch mais notre travail d'utilisation de la librairie Skorch<sup>5</sup> pour exploiter les fonctionnalités de Pytorch sous Scikit-Learn n'a pas abouti.

Nous avons donc choisi d'utiliser un autre réseau de neurones : le MLP (Multilayer perceptron), disponible sur Scikit-Learn mais non récurrent. Le MLP est un réseau de neurones artificiels à action directe (i.e. le chemin du signal à travers les nœuds ne va que dans un sens) qui utilise la *rétropropagation du gradient* pour former le réseau. À l'exception des neurones d'entrée, chaque neurone utilise une fonction d'activation non linéaire : ainsi le MLP peut distinguer des données qui ne sont pas linéairement séparables.

Nous désirions aussi appliquer le MDM[2] (minimum distance to mean classifier). Cet algorithme calcule la distance euclidienne de chaque pixel inconnu au vecteur moyen de chaque classe. Tous les pixels sont classés dans la classe la plus proche, à moins qu'un écart-type ou un seuil de distance ne soit spécifié, auquel cas certains pixels peuvent être non classés. Cependant, le MDM est disponible sur pyRiemann mais pas sur Scikit-Learn. De plus, les données d'entrée de cet algorithme ne sont pas les données brutes, comme c'est le cas pour les algorithmes déjà sélectionnés, mais la covariance des données. Pour ces deux raisons, nous avons choisi de l'écarter.

Ainsi nous travaillerons avec Scikit-Learn en utilisant les algorithmes KNN, Random Forest et MLP pour ce projet.

5. Lien vers la librairie skorch

### 3.2 Traitement des données et calculs

Après avoir pris en main la librairie Scikit-Learn ainsi que les données Breizhcrops, nous nous sommes lancés dans la création d'une *pipeline de classification* (fig. 3).

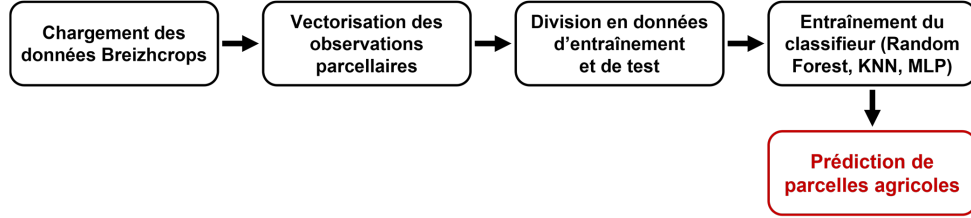


FIGURE 3 – Chaîne de traitement de notre pipeline de classification

Le coeur de notre programme repose sur le fichier `train_test_split.py`, une variante de la fonction Scikit-learn qui permet la division du jeu de données brut en données d'entraînement d'une part, et en données de test d'autre part. Au lancement de ce script, l'utilisateur est invité à saisir le nombre et les couples (*region, annee*) souhaités pour la partition des données, et renvoie en sortie un 4-uplet  $(X_{train}, X_{test}, y_{train}, y_{test})$  correspondant respectivement aux observations d'entraînement, observations de test, classes d'entraînement, classes de test. En interne, le programme récupère les vecteurs  $X$  (observations) et  $y$  (classes associées) issue d'une *vectorisation* effectuée en amont, dans des fichiers sauvegardés au préalable. En effet, les fonctions de classification proposées par Scikit-Learn nécessitent que le vecteur des observations soit de dimension 2 (*nombre d'observations \* nombre d'attributs*). Ce processus est illustré par la figure 4 :

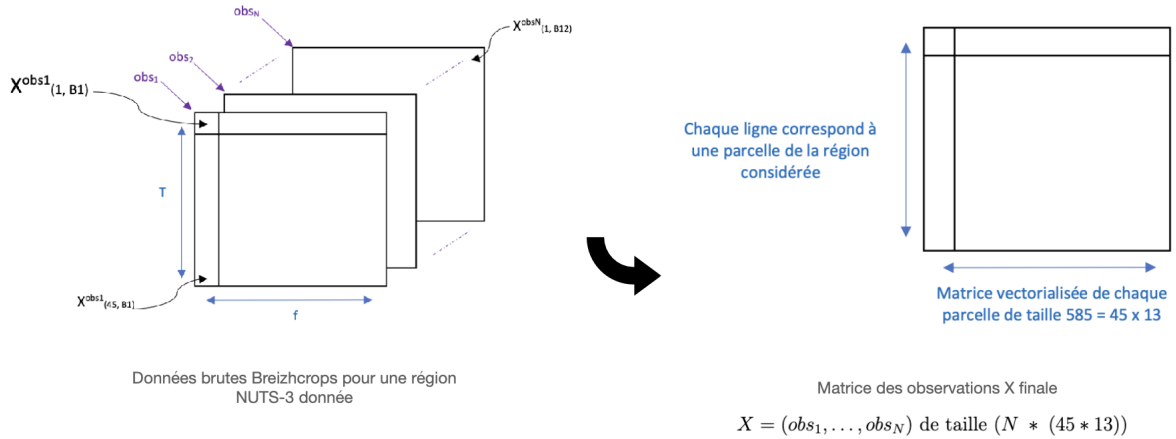


FIGURE 4 – Vectorisation des observations parcellaires

avec : pour  $N$  le nombre de parcelles,  $\forall i \in \{1, \dots, N\}$ ,  $X^{obs_i}$  Observation pour la parcelle  $i$   
 $f \in \{B1, \dots, B12\}$  Bandes de fréquence de Sentinel-2  
 $T \in \{1, \dots, 45\}$  Répartition du temps sur une année en 45 dates

Il s'en suit alors la phase d'entraînement du classifieur choisi (Random Forest, KNN, MLP) sur les données d'entraînement, puis la classification sur les données de test.



Pour la qualification de la prédiction, nous choisissons de calculer l'*accuracy*, la *matrice de confusion*, ainsi que le *F-score*, sans oublier le *temps de calcul* mis pour chacun des algorithmes.

## 4 Résultats et discussion

### 4.1 Étude de la classification

Pour les tests, la partition classique a été adoptée pour les données :

- 75% en données d'entraînement
- 25% en données de test

En jouant sur la distribution des données, tout en fixant les différents paramètres des classifieurs à leurs valeurs par défaut, nous établissons les tables 1 et 2 :

	Random Forest	KNN (3 voisins)	MLP
Accuracy (%)	46.3	26.3	<b>74.2</b>
F-score (%)	35.3	26.9	73.3
Temps de calcul (s)	6	15152 ( $\approx$ 4 h 13 min)	690 ( $\approx$ <b>12 min</b> )

TABLE 1 – Performances moyennes sur 3 itérations avec les données 2017 : *entraînements sur* (FRH01, FRH02, FRH03) *et tests sur* FRH04 (cf. Annexe pour les matrices de confusion)

	Random Forest	KNN (3 voisins)	MLP
Accuracy (%)	44	25.8	<b>74.1</b>
F-score (%)	32.4	25.9	73.7
Temps de calcul (s)	5	2692 ( $\approx$ 45 min)	854 ( $\approx$ <b>15 min</b> )

TABLE 2 – Performances moyennes sur 3 itérations avec les données 2017 : *entraînements sur* (FRH02, FRH03, FRH04) *et tests sur* FRH01 (cf. Annexe pour les matrices de confusion)

À première vue, les résultats des deux configurations semblent assez proches : sachant que la taille du jeu de test diffère de 60000 parcelles environ entre les 2 TABLES (tests sur le plus grand et le plus petit département de l'ensemble des données), cela reste du même ordre de grandeur.

On remarque alors que le **MLP** fournit des résultats très satisfaisants par rapport à ses deux concurrents, avec une précision moyenne légèrement inférieure à 75%, i.e. *plus de 3 parcelles sur 4 sont correctement classées*. Le temps de calcul reste aussi relativement correct : si l'on compare avec le KNN, celui-ci met dans le meilleur des cas 3 fois plus de temps, pour donner un résultat beaucoup moins bon.

Le **Random Forest** quant à lui donne une précision plus satisfaisante, puisqu'il dépasse le KNN de près de 20% sur cette statistique, et un temps de calcul quasi-instantané.

Compte-tenu des **scores très faibles obtenus avec le KNN**, nous avons cherché à étudier l'influence du nombre de voisins sur ses performances (voir FIGURE 5).

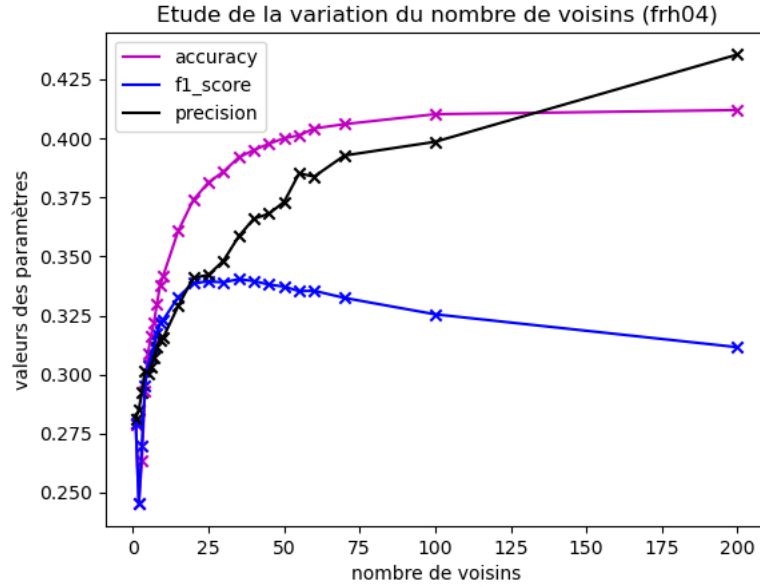


FIGURE 5 – Valeurs moyennes sur 3 itérations des paramètres statistiques en fonction du nombre de voisins de l’algorithme KNN (données 2017, *entraînements* sur (*frh01*, *frh02*, *frh03*) et *tests* sur *frh04*), poids et distance par défaut

Le graphique en FIGURE 5 montre l’évolution du score de classification en fonction du nombre de voisins  $k$ . On remarque que le taux global de bonnes classification (*accuracy*) et le taux de classification par classe (*precision*) augmentent avec  $k$ , montrant qu’il existe une autocorrélation spatiale des parcelles agricoles. Malgré cet effet, ces deux métriques atteignent un plateau lorsque  $k$  est grand (plus marqué pour l’*accuracy* que pour la *précision*). Cela montre qu’il existe en moyenne une relative homogénéité locale des cultures. L’augmentation de la *précision* peut s’expliquer par un effet d’agrégation de certaines classes (cultures), d’où l’utilité d’incorporer plus de voisins (parcelles) dans le classifieur.

Nous nous attendions également à observer une diminution de la *précision* à partir d’une certaine augmentation de  $k$  : puisque les champs trop éloignés (et donc d’agrégation différente) pourraient induire en erreur le KNN. Le temps de calcul étant extrêmement long, il est fastidieux d’obtenir davantage de résultats en continuant d’augmenter le nombre de voisins  $k$  sur la région *frh04*. Nous avons donc réalisés les mêmes tests mais sur la région Belle-Île qui possède 100 fois moins de parcelles : le temps de calcul est passé de 40 min à 20 secondes. Nous avons ainsi pu faire varier le nombre de voisins de 1 à 1000 et observer les mêmes phénomènes que précédemment c’est-à-dire que l’*accuracy* et le *f-score* tendent vers une valeur constante tandis que la *précision* augmente lorsque  $k$  croît. Néanmoins, au bout de 400 voisins, la *précision* atteint elle aussi un seuil avant de chuter brutalement à environ 700 voisins.

Nous avons également étudié l’évolution de ces métriques et du temps de calcul en fonction des 3 paramètres du KNN : c’est-à-dire le nombre de voisins, le poids et la distance. Cette étude a été effectuée sur la région Belle-Île.

## 4.2 Extension du jeu de données

On sait désormais comment étendre le jeu de données de façon temporelle mais nous avons manqué de temps pour produire cette extension. Il est donc possible d'obtenir des données parcellaires sur la période 2018 à 2022 et l'ensemble de nos recherches et de nos résultats sont accessibles librement en ligne sur notre repository Github<sup>6</sup>.

## 5 Conclusion et perspectives

Notre étude a permis plusieurs avancées vis-à-vis de la manipulation du jeu de données Breizhcrops.

Nous avons, tout d'abord, mis en évidence les moyens de récupérer les données des parcelles pour une année souhaitée. Nous avons également explicité la chaîne de traitement nécessaire afin de réutiliser facilement différents algorithmes de classification dans le but d'une étude multi-temporelle des parcelles agricoles grâce à la librairie Scikit-Learn.

Cependant, nous n'avons pas réussi à mettre en place de classification multi-temporelle en raison du manque de temps pour produire cette extension du jeu de données.

### Les difficultés rencontrées

Effectivement, tout au long du projet, notre équipe a été confrontée à différents obstacles qui nous ralentis par rapport à l'objectif que nous nous étions fixé au départ :

- Un manque d'informations sur la prise en main du jeu de données Breizhcrops, même pour un public averti.
- Les données Breizhcrops utilisent le module Pytorch qui nécessite une compréhension solide en Machine Learning (contrairement à Scikit-Learn).
- Le volume des données étant très important, il a été nécessaire de les sauvegarder en amont du calcul de classification.
- Les calculs sont parfois très longs selon le jeu de données, le classifieur et la machine utilisée.

### Perspectives et améliorations à faire

Notre projet s'achève, mais il constitue désormais un socle pour de futurs projets. Voici quelques pistes possibles d'amélioration :

- Poursuivre nos recherches en récupérant les données parcellaires post-2017 et les croiser dans la composition du jeu d'entraînement et jeu de test (évolution de la végétation au cours du temps).
- Généralisation de la classification de parcelles agricoles sur d'autres régions françaises.
- Prendre en compte des attributs spatiaux, en plus de la réflectance, comme des informations sur la géométrie et la topologie des parcelles.

---

6. <https://github.com/ar8n/PIR-Breizhcrops>

## 6 Annexes

### Glossaire

**Accuracy** : rapport du nombre de parcelles bien classées sur le nombre total de parcelles.

**Benchmark** : moyens permettant d'évaluer les performances de la classification.

**Culture principale** : si plusieurs plantes sont cultivées successivement sur une même parcelle, la culture principale correspond à la plante cultivée ayant occupé le sol le plus longtemps et couvrant la majorité de la parcelle.

**Complexité** : quantité de ressources (en temps et en mémoire) nécessaire à l'exécution d'un algorithme.

**Forêt aléatoire (Random Forest)** : algorithme d'apprentissage automatique effectuant un apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles de données différents.

**F-score** : moyenne harmonique pondérée entre la précision et le rappel. Sa meilleure valeur est 1 et sa pire valeur est zéro.

**frh0x avec  $x \in \{1, 2, 3, 4\}$**  : départements de la région Bretagne utilisés dans notre étude sous nomenclature NUTS-3 (Nomenclature des unités territoriales statistiques, standardisé au sein de l'Union Européenne)

**Pipeline de classification** : chaîne de traitement, de l'exploration des données à la classification, cela comprend à la fois le traitement des données, les modèles et leur évaluation.

**Précision** : pour une classe donnée, rapport du nombre de parcelles bien classés sur le nombre total de parcelles affectés à cette classe.

**Rappel** : capacité pour le classifieur à trouver tous les exemples positifs.

**Rétropropagation du gradient** : méthode pour entraîner un réseau de neurones, consistant à mettre à jour les poids de chaque neurone de la dernière couche vers la première (visant à corriger les erreurs selon l'importance de la contribution de chaque élément à celles-ci).

**Surapprentissage** : on parle de surapprentissage lorsqu'un modèle apprend des descripteurs spécifiques à un ensemble de données d'entraînement l'empêchant de généraliser sur d'autres ensembles de données.

## Principe général de la classification

La classification, en télédétection, a pour objectif de regrouper des objets spatiaux en un certain nombre de classes les plus homogènes possibles. Il existe deux principaux types de classification :

- la **classification non supervisée**, parfois appelée partitionnement, segmentation ou regroupement (*clustering* en anglais) qui est utilisée lorsqu'on ne connaît pas le nombre de groupes (ou classes) qui existent pour répartir les observations à classer et/ou lorsque l'on ne connaît pas le groupe auquel appartient chaque observation.
- la **classification supervisée**, souvent appelée simplement classification (*classification* en anglais) est utilisée lorsque l'on connaît déjà le nombre de groupes qui existent et le groupe auquel appartient chaque observation que l'on veut classer. Cette seconde classification se fait en deux étapes distinctes : une phase d'entraînement et une phase de test (ou de généralisation).

## Matrice de confusion

Une matrice de confusion aide à synthétiser le résultat d'une classification. Elle est de dimension  $(nb_{classes} \times nb_{classes})$ , et permet de répertorier le nombre de parcelles ayant été affecté à la bonne classe ou à la mauvaise classe. Ainsi, l'élément en position  $(i, j)$  se traduit par le nombre de parcelles ayant été prédite dans la classe  $i$  mais étant réellement dans la classe  $j$ . La figure 6 donne un exemple de ce genre de matrice dans le cas d'une classification binaire :

		Reality	
Confusion matrix		Negative : 0	Positive : 1
Prediction	Negative : 0	True Negative : TN	False Negative : FN
	Positive : 1	False Positive : FP	True Positive : TP

FIGURE 6 – Matrice de confusion pour une classification binaire<sup>7</sup>

Ci-dessous les matrices de confusion associées aux tables 1 et 2 :

	0	1	2	3	4	5	6	7	8
0	0	44.3333	0	1105	0	0	0	0.333333	4828.33
1	0	1081	0	4127	0	0	0	4.33333	11780.7
2	0	18	0	754	0	0	0	0	2464
3	0	15.6667	0	22924.7	0	0	0	1	8391.67
4	0	0	0	0.666667	0	0	0	0	1.33333
5	0	1	0	125	0	0	0	0	427
6	0	0	0	3.66667	0	0	0	0	7.33333
7	0	32	0	4197	0	0	0	3	21885
8	0	41	0	5552.67	0	0	0	1.33333	32796

(a) Random Forest

	0	1	2	3	4	5	6	7	8
0	895	1611	168	985	0	9	0	625	1685
1	2448	5051	462	2976	1	40	0	1702	4313
2	428	836	151	617	0	7	0	318	879
3	3339	6330	878	11085	1	46	1	2948	6705
4	0	1	0	1	0	0	0	0	0
5	67	125	15	114	0	0	0	70	162
6	1	5	1	1	0	0	0	1	2
7	3320	5723	732	5368	0	64	0	3295	7615
8	4824	8729	1132	7409	0	112	0	4396	11789

(b) KNN (3 voisins)

	0	1	2	3	4	5	6	7	8
0	4258.67	893.667	132	162.667	0	0	0	159.333	371.667
1	689.333	15344	45	236.333	0	0	0	226	452.333
2	76	26	2932.33	49.3333	0	0	0	38	114.333
3	84	177.667	55.3333	30104.7	1.66667	0	1	215.333	693.333
4	0	0	0	0	0	0	0	0	2
5	6	18.6667	5.66667	36.6667	0	0	0	239.667	246.333
6	0.333333	0	0	1	0	0	0	6	3.66667
7	264.667	402.667	35	615.333	0	0	5	10832.3	13962
8	729	634.667	189.333	1338	0	0.333333	1.66667	8019.67	27478.3

(c) MLP

FIGURE 7 – Matrices de confusion pour la TABLE 1

	0	1	2	3	4	5	6	7	8
0	0	46.3333	0	2002.33	0	0	0	2	10995.3
1	0	814	0	5411.33	0	0	0	11.6667	24131
2	0	41.6667	0	1490.33	0	0	0	1.66667	4059.33
3	0	1.66667	0	31130.7	0	0	0	6	12851.7
4	0	0	0	0.333333	0	0	0	0	0.666667
5	0	0	0	172	0	0	0	0	772
6	0	0	0	3.33333	0	0	0	0	6.66667
7	0	8	0	4107.33	0	0	0	22.6667	28512
8	0	13.6667	0	5360	0	0	0	17.3333	46620

(a) Random Forest

	0	1	2	3	4	5	6	7	8
0	1692	2977	365	2513	0	31	0	1667	3801
1	4003	8179	811	5607	0	78	0	3392	8298
2	663	1388	237	1032	0	16	0	642	1615
3	4330	8436	1069	14707	0	110	0	4828	10510
4	0	0	0	1	0	0	0	0	0
5	100	230	28	222	0	0	0	98	266
6	4	1	0	2	0	0	0	0	3
7	3676	6423	855	7088	1	94	0	4582	9931
8	5838	9887	1387	10658	0	135	2	7384	16720

(b) KNN (3 voisins)

	0	1	2	3	4	5	6	7	8
0	7702.67	3222.67	115	193	0.666667	0	0	349.667	1462.33
1	1216.33	27118.3	32	274.333	0	0	0	436.667	1290.33
2	201.333	96.6667	4628.33	54.3333	0.333333	0	0	53.6667	558.333
3	106.667	305	54.6667	40981.7	1.66667	0	1.66667	650.667	1888
4	0	0	0	0	0	0	0	0	1
5	12	40.6667	3	52.6667	0.333333	0	0.333333	291.333	543.667
6	0	0	0	0	0	0	0	7.66667	2.33333
7	158	508.667	41.6667	567	1.33333	0	1.33333	13911.7	17460.3
8	414.667	723.667	81.3333	1013.67	1.33333	0	1.33333	11680.3	38094.7

(c) MLP

FIGURE 8 – Matrices de confusion pour la TABLE 2

## Références

- [1] Earth Engine. A planetary-scale platform for Earth science data and analysis.
- [2] M. Congedo C. Jutten A. Barachant, S. Bonnet. Multiclass brain–computer interface classification by riemannian geometry, 2012.
- [3] Léo Breiman. Random forests, 2001.
- [4] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1) :21–27, 1967.
- [5] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn : Machine learning in python. *the Journal of machine Learning research*, 12 :2825–2830, 2011.
- [6] Félix Quinton. Suivi de la rotation des cultures à partir de séries temporelles d’images satellite, 2021.
- [7] Marc Rußwurm, Charlotte Pelletier, Maximilian Zollner, Sébastien Lefèvre, and Marco Körner. Breizhcrops : A time series dataset for crop type mapping. *arXiv preprint arXiv :1905.11893*, 2019.