REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ DES MATHEMATIQUES ET D'INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : Réseaux et Télécommunications

Par :

Said Taha Rami
Guendouz Ahmed Ilyes

Sur le thème

Modélisation et Implémentation d'un Système de Prédiction d'Arrosage Automatique des Plantes à l'aide de l'Apprentissage Automatique

Soutenu publiquement le 13/ 06 / 2024 à Tiaret devant le jury composé de :

Mr MOSTEFAOUI Sid Ahmed Mokhtar	MCA	Université Tiaret	Président
Mr MEGHAZI Hadj Madani	MCB	Université Tiaret	Encadrant
Mr DAOUD Mohamed Amine	MCB	Université Tiaret	Examinateur

2023-2024

Dédicaces

Je dédie ce projet de fin d'études à mes parents, à mes frères et sœurs, à mes amis, ainsi qu'à mes neveux et nièces : Ghoufran, Yasser, et Asser El-Djilali. Votre soutien inébranlable m'a permis de mener à bien ce travail.

SAID Taha Rami.

Dédicaces

Je dédie ce projet de fin d'études à ma mère, mes sœurs Ikhlasse et Douaa, mon frère Amine, à mes amis, et à toute ma famille, piliers de soutien et d’encouragement tout au long de ce parcours, tant moralement que matériellement. Une pensée particulière pour mon père Djamel Eddine Rabi Yrahmou.

GUENDOUZ Ahmed Ilyes.

REMERCIEMENT

Louange à Allah, le Tout-Puissant et Miséricordieux, qui m'a accordé la sagesse et la santé nécessaires pour mener à bien ce travail malgré les défis rencontrés.

Je suis profondément reconnaissant envers l'encadrant M. **MEGHAZI Hadj Madani** pour son soutien indéfectible, ses précieux conseils et sa confiance en mes capacités. Sa disponibilité et ses encouragements constants ont été une source d'inspiration et m'ont permis de progresser avec confiance.

Mes sincères remerciements vont également à M. **MOSTEFAOUI Sid Ahmed Mokhtar** et M. **DAOUD Mohamed Amine** pour avoir accepté de faire partie du jury de ce mémoire. Votre expertise et vos commentaires constructifs sont grandement appréciés.

Un grand merci aux professeurs et à l'administration de la faculté d'informatique et de mathématiques pour leur encadrement et leur soutien tout au long de mes études.

Enfin, je remercie tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail par leurs conseils, leurs encouragements et leurs commentaires constructifs.

SAID Taha Rami.
GUENDOUZ Ahmed Ilyes.

RÉSUMÉ

L'application du Machine Learning dans l'agriculture de précision offre des perspectives prometteuses pour optimiser diverses facettes de l'agriculture, telles que l'irrigation, la fertilisation et la gestion des cultures. Dans la littérature, de nombreux travaux de recherche ont abordé l'utilisation des modèles de Machine Learning pour améliorer ces aspects agricoles, mais les approches traditionnelles manquent souvent de précision et de robustesse. Ce travail présente un modèle innovant d'irrigation automatique en utilisant la méthode de Stacking d'Ensemble Learning. Notre approche combine les forces des modèles LightGBM, XGBoost et Support Vector Machine pour atteindre des performances supérieures. Les résultats montrent une amélioration significative en termes de précision et de fiabilité par rapport aux modèles de Machine Learning de base et aux autres modèles d'Ensemble Learning testés. Cette avancée offre une solution plus robuste et efficace pour l'irrigation automatique, posant les bases pour de futures recherches et applications dans le domaine de l'agriculture de précision. Ce travail démontre le potentiel de l'Ensemble Learning pour résoudre des problèmes complexes et améliorer les pratiques agricoles.

Mots-clés

Irrigation automatique, Agriculture de précision, Machine Learning, Stacking, Ensemble Learning, LightGBM, XGBoost, Support Vector Machine.

ABSTRACT

The application of Machine Learning in precision agriculture holds promising prospects for optimizing various facets of farming, including irrigation, fertilization, and crop management. While existing research has explored the use of Machine Learning models to enhance these agricultural aspects, traditional approaches often lack precision and robustness. This paper introduces an innovative model for automatic irrigation using the Stacking Ensemble Learning method. Our approach combines the strengths of LightGBM, XGBoost, and Support Vector Machine models to achieve superior performance. Results demonstrate a significant improvement in terms of precision and reliability compared to basic Machine Learning models and other tested Ensemble Learning models. This advancement provides a more robust and efficient solution for automatic irrigation, laying the groundwork for future research and applications in precision agriculture. This work underscores the potential of Ensemble Learning to address complex challenges and enhance agricultural practices.

Keywords

Automatic irrigation, Precision agriculture, Machine Learning, Stacking, Ensemble Learning, LightGBM, XGBoost, Support Vector Machine.

ملخص

يوفر تطبيق التعلم الآلي في الزراعة الدقيقة آفاقًا واعدة لتحسين مختلف جوانب الزراعة، مثل الري والتسميد وإدارة المحاصيل. وقد تناولت العديد من الأعمال البحثية في الأدبيات استخدام نماذج التعلم الآلي لتحسين هذه الجوانب الزراعية، ولكن غالبًا ما تفتقر المناهج التقليدية إلى الدقة والمتانة. يقدم هذا العمل نموذجًا مبتكرًا للري الآلي باستخدام طريقة تكديس التعلم التجميعي. ويجمع نهجنا بين نقاط القوة في نماذج LightGBM و XGBoostو Support Vector Machine لتحقيق أداء متفوق. تظهر النتائج تحسنًا كبيرًا في الدقة والموثوقية مقارنة بنماذج التعلم الآلي الأساسية ونماذج التعلم التجميعي الأخرى التي تم اختبارها. ويوفر هذا التقدم حلاً أكثر قوة وفعالية للري الآلي، مما يضع الأسس للأبحاث والتطبيقات المستقبلية في مجال الزراعة الدقيقة. يوضح هذا العمل إمكانات التعلم التجميعي لحل المشاكل المعقدة وتحسين الممارسات الزراعية.

الكلمات المفتاحية

الري الأوتوماتيكي، الزراعة الدقيقة، التعلم الآلي، التكديس، التعلم التجميعي، LightGBM، XGBoost، آلة دعم المتجهات.

Table des matières

Liste des abréviations

AP	Agriculture de Précision
GPS	Global Positioning Systems
DGPS	Differential Global Positioning Systems
SGSC	Systèmes de Gestion des Sols et des Cultures
IoT	Internet of Thing
ML	Machine Learning
DL	Deep Learning
IA	Intelligence artificial
SIG	Geographic Information Systems
NDVI	Normalized Difference Vegetation Index.
NIR	Near InfraRed
UAV	Unmanned Aerial Vehicle
KNN	K-Nearest Neighbors
SVM	Support Vector Machine
ANN	Artificial neural network
RF	Random Forest
GBM	Gradient Boosting
CNN	Convolutional Neural Network
RMSE	Root Mean Square Error
ELM	Extreme Learning Machines

Listes des figures

Liste des tableaux

Introduction générale

Introduction Générale

L’agriculture est un pilier fondamental du développement humain, ayant évolué pour répondre aux besoins croissants d'une population mondiale en constante augmentation. Avec une population projetée à 9 milliards d’individus d’ici 2050, il est impératif d’augmenter la production agricole de 60% par rapport aux niveaux actuels pour satisfaire la demande alimentaire mondiale. Cette nécessité est augmentée par les défis supplémentaires posés par les changements climatiques, le manque d'eau et les effets environnementaux négatifs liés à l'utilisation intensive de fertilisants.

L’agriculture de précision émerge comme une solution prometteuse pour relever ces défis. Ce concept repose sur l’utilisation de technologies avancées pour optimiser l’utilisation des ressources agricoles, réduire les pertes et maximiser les rendements.

A travers ce projet, nous essayons de traiter le problème de commande automatique d’irrigation des plantes et faire l’investigation au près du domaine du ML pour trouver des solutions adéquates qui permettent une exploitation idéale des ressources, plus précisément en eau.

Pour faire face à cette problématique, nous avons adopter le plan de travail suivant :

L**e Chapitre 1**, offre un aperçu complet de l'agriculture de précision, couvrant ses définitions, ses origines historiques, et les motivations qui ont conduit à son développement. Nous explorerons également les applications concrètes de ces techniques, notamment dans la gestion des cultures et de l'irrigation, tout en analysant les défis qui freinent leur adoption généralisée. Les technologies clés qui rendent possible l’agriculture de précision, telles que les drones, les systèmes de géolocalisation, les capteurs, et l'imagerie satellitaire.

Le Chapitre 2 traite le rôle crucial de l'Intelligence Artificielle (IA) et du Machine Learning (ML) dans la révolution technologique actuelle. Nous essayons de décrypter ces concepts en clarifiant leurs liens et distinctions avant de plonger dans les différentes approches de ML, y compris les techniques supervisées, non supervisées, semi- supervisées et par renforcement. Nous explorerons également les algorithmes qui animent le ML, des méthodes classiques comme la régression linéaire aux réseaux de neurones avancés, en passant par les techniques d'Ensemble Learning et de Deep Learning.

Dans **le Chapitre 3**, nous passerons en revue les travaux récents sur l'irrigation utilisant le Machine Learning. Cette section examinera les études et les innovations qui ont permis d'améliorer l'efficacité de l'irrigation à travers des modèles prédictifs et des systèmes intelligents.

Le Chapitre 4 se concentrera sur le développement et la mise en œuvre d'un système d'irrigation intelligent. Nous décrirons en détail chaque composant du système, depuis les capteurs jusqu'à l'interface utilisateur, en passant par le processus de mise en place de la solution d'irrigation et l'entraînement des modèles d'apprentissage automatique. Nous présenterons également les résultats de nos tests de divers modèles de ML, incluant des techniques d'apprentissage de base ainsi que des méthodes d'Ensemble Learning comme le Boosting et le Stacking.

Une conclusion générale vient clore ce travail en donnant un résumé, des conclusions et en suggérant de nouvelles perspectives pour les recherches futures dans ce domaine.

CHAPITRE I
L'Agriculture de Précision

Introduction

Ce chapitre offre un aperçu complet de l'agriculture de précision, une approche novatrice qui révolutionne les pratiques agricoles. Nous aborderons dans un premier temps les définitions et les origines historiques de ce concept. Quels sont les principes fondateurs et les principales motivations qui ont conduit à l'émergence de l'agriculture de précision ?

Nous étudierons ensuite les nombreuses applications concrètes de ces techniques de pointe dans la gestion des cultures et de l'irrigation. Comment l'agriculture de précision permet-elle d'optimiser de manière fine l'apport en intrants, la fertilisation ou encore le pilotage de l'irrigation au sein d'une même parcelle ?

Malgré ses avantages avérés, le déploiement de l'agriculture de précision soulève certains défis qu’il sera essentiel d'analyser. Nous identifierons les principaux freins actuels à une adoption généralisée de ces pratiques.

Enfin, nous examinerons en détail les différentes technologies clés qui rendent possible l'agriculture de précision, des drones et systèmes de géolocalisation aux capteurs de sol haute résolution, en passant par l'imagerie satellitaire et les systèmes d'information géographique. Nous verrons comment ces outils permettent une caractérisation fine des variabilités intra-parcellaires.

L'objectif de ce chapitre est de fournir une vision d'ensemble de l'agriculture de précision, en couvrant à la fois les fondements conceptuels, les défis, les applications concrètes ainsi que les technologies innovantes impliquées.

1. L'agriculture de précision

L'agriculture de précision (AP) représente une gamme de méthodes et de stratégies qui cherchent à réduire l'emploi direct d'intrants dans les champs agricoles. En prenant en considération la variabilité des conditions à l'intérieur même des parcelles, cette approche permet de personnaliser l'application des intrants en ajustant leur quantité, leur moment et leur emplacement de manière précise, dans le but d'optimiser les résultats économiques, agronomiques et environnementaux des exploitations agricoles [1].

Les agriculteurs tirent parti de technologies telles que le GPS, les drones et les images satellites pour bénéficier d'informations cruciales concernant divers aspects de leur exploitation, notamment l'état des cultures, les prévisions météorologiques et les changements environnementaux [2].

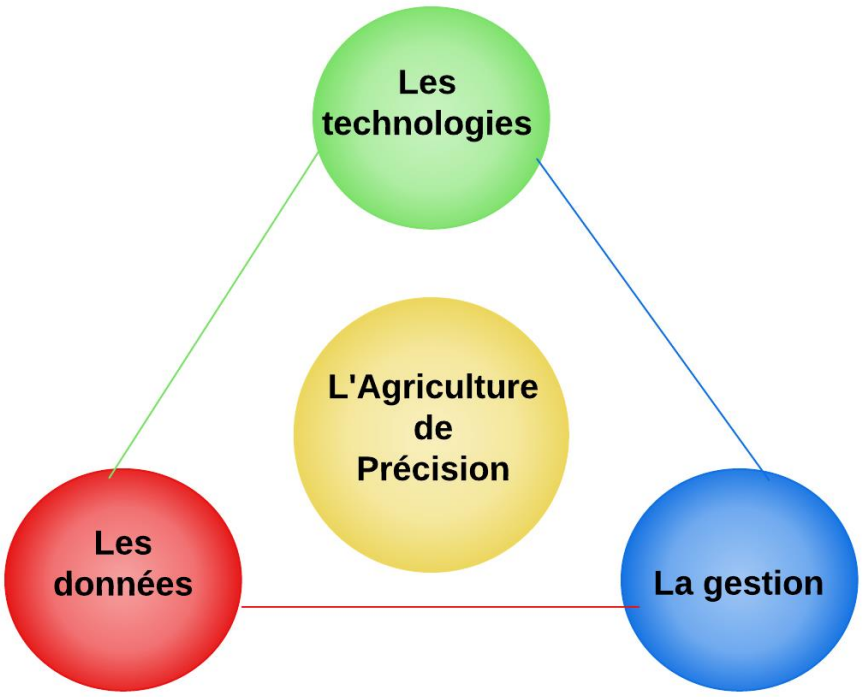


Figure I.1 Composantes majeures de l'agriculture de précision.

1.1 Historique

L'agriculture de précision est une approche de gestion agricole qui utilise les technologies de l'information pour optimiser la production en prenant des décisions basées sur des données provenant de différentes sources. Elle est adoptée dans divers pays, des grandes exploitations aux petites. L'évolution de la L'agriculture de précision est influencée par les progrès

technologiques, les besoins sociaux et environnementaux. Initialement axée sur la variabilité des sols, elle s'est développée avec l'avènement de l'électronique et des technologies de l'information. L'adoption du GPS en 1995 a marqué une étape cruciale, permettant une collecte de données géo-référencées plus précise. Bien que désignée sous différents noms au début, le terme "agriculture de précision" est devenu le plus couramment utilisé depuis les années 1990 [3].

La chronologie du développement de l'agriculture de précision met en lumière les principales étapes de l'évolution des pratiques agricoles vers des méthodes plus précises et efficaces. Voici une explication plus détaillée de chaque étape [4] :

- **Fin du 19ème siècle :** Cette période marque le début de la compréhension de la variation spatiale dans le sol et l'environnement. Les premiers agriculteurs ont observé que les propriétés du sol, le microclimat, le drainage, les caractéristiques du paysage et l'utilisation précédente des terres contribuaient tous à la variabilité au sein des champs.

- **Milieu du 20ème siècle :** Les chercheurs agricoles comme Gilbert et Lawes à la Rothamsted Research au Royaume-Uni ont commencé à entreprendre des efforts pour évaluer et gérer cette variation au sein des champs. Leur travail était axé sur l'expérimentation de différentes combinaisons et quantités de nutriments pour les cultures et de variétés pour augmenter les rendements.

- **Années 1970** : Les agriculteurs ont commencé à reconnaître les avantages potentiels d'une meilleure tenue de registres et de compréhension des besoins en intrants du sol et des cultures. Cette prise de conscience accrue a posé les bases pour des pratiques de gestion plus précises.

- **Années 1980** : Les progrès technologiques, notamment dans les machines agricoles, ont incité à passer d'une gestion des fermes dans leur ensemble à une gestion de la variation au sein des champs. Les agriculteurs ont commencé à fusionner de plus petits champs en unités plus grandes pour accommoder des machines plus grandes, ce qui a entraîné une augmentation de la variabilité au sein des champs.

L'introduction de compteurs de rendement par des sociétés comme Massey Ferguson a permis l'enregistrement continu des rendements des cultures. Cela a fourni aux agriculteurs des informations précieuses sur les variations de rendement au sein des champs.

- **Années 1990** : Les systèmes de positionnement global (GPS) sont devenus disponibles sur les tracteurs, permettant la cartographie régulière des rendements. Cette technologie a fourni aux agriculteurs des données spatiales précises pour de meilleures décisions de gestion.

- **Début des années 2000** : La technologie du GPS différentiel (DGPS) a considérablement amélioré la précision du GPS, rendant possible une précision submétrique. Cette précision a permis une cartographie et une gestion plus détaillées des pratiques agricoles.

- **Milieu des années 2000** : Le développement et l'adoption de la technologie d'application à taux variable (VRT) ont permis aux agriculteurs d'appliquer des engrais et des pesticides à des taux variables dans les champs, sur la base de données en temps réel et de la variabilité spatiale.

1.2 Découvrir les Bénéfices et l'Impact de l'Agriculture de Précision

L'intégration des principes de l'agriculture de précision entraîne une amélioration significative de l'efficacité des cultures et une réduction des coûts financiers, tout en augmentant la production. Bien que les technologies puissent sembler coûteuses à première vue, à long terme, elles génèrent des économies plus importantes par rapport aux méthodes traditionnelles. Par exemple, les agriculteurs peuvent appliquer avec précision la quantité d'engrais nécessaire et déterminer les types les plus efficaces pour des zones spécifiques. De plus, ces technologies permettent une planification agricole améliorée sur le long terme, en ajustant les stratégies en temps réel pour faire face à des événements imprévus [2].

- Optimiser l'utilisation des ressources naturelles telles que l'eau, le sol et les plantes pour accroître leur efficacité.
- Diminuer la nécessité d'utiliser une gamme de produits agrochimiques comme les pesticides et les engrais grâce à des pratiques plus ciblées.
- Réduire les intrants tout en améliorant la rentabilité des opérations agricoles.

- Renforcer la sécurité alimentaire en garantissant des rendements agricoles plus stables et fiables.
- Contribuer à la préservation de l'environnement en limitant les impacts négatifs de l'agriculture sur les écosystèmes locaux [5].

1.3 Les défis de l'agriculture de précision

L'agriculture de précision vise à optimiser les rendements agricoles tout en réduisant l'utilisation d'intrants coûteux et potentiellement dommageables pour l'environnement. Grâce aux technologies numériques, cette approche permet de prendre des décisions plus éclairées basées sur des données précises. Cependant, son adoption généralisée fait face à plusieurs obstacles. Il y a quatre défis majeurs :

1.3.1 Le coût élevé des technologies

Mettre en place les technologies de l'agriculture de précision représente un investissement considérable. Le coût d'acquisition des drones, des capteurs de sol et météorologiques, des systèmes de guidage GPS, des logiciels de cartographie et d'analyse des rendements peut facilement atteindre des dizaines de milliers d'euros. S'ajoutent à cela les coûts récurrents liés à la maintenance, aux mises à jour, aux abonnements de services en ligne et à la formation continue du personnel. Pour de nombreuses exploitations agricoles, en particulier les petites structures ou celles disposant de ressources financières limitées, ces dépenses peuvent être insurmontables et freiner l'adoption de ces technologies pourtant prometteuses.

1.3.2 La connectivité et l'infrastructure

L'agriculture de précision moderne repose sur des échanges permanents de données entre le terrain et le cloud/centres de traitement. Une connexion Internet haut débit fiable et stable est donc indispensable pour transférer en temps réel les données des capteurs, drones, etc. Or, de nombreuses zones rurales souffrent encore d'une couverture réseau limitée voire inexistante, entravant l'utilisation optimale de ces technologies. De même, une infrastructure informatique performante (serveurs, cyber-sécurité, etc.) est nécessaire pour héberger et traiter en toute sécurité ces masses de données sensibles. Son déploiement représente un défi logistique et financier de taille.

1.3.3 La complexité des données et leur gestion

L'agriculture de précision repose sur la collecte et l'analyse d'un volume massif de données provenant de multiples sources : images satellites haute résolution, données des capteurs IoT déployés dans les champs, relevés météorologiques locaux, historiques de rendements, etc. Combiner, structurer et traiter ces données hétérogènes issues de différents formats pose un réel défi technique. Il nécessite des compétences pointues en sciences des données, en analyse d'images et en modélisation prédictive que la plupart des agriculteurs ne possèdent pas. De plus, le stockage sécurisé de ces précieuses données agronomiques et leur protection contre le piratage représentent des enjeux majeurs [6].

1.3.4 La formation et les compétences

Pour tirer pleinement parti des technologies de l'agriculture de précision, une solide formation des agriculteurs et de leurs équipes est indispensable. Bien au-delà de l'utilisation des outils eux-mêmes, il faut acquérir une compréhension approfondie des concepts et méthodologies sous-jacents : échantillonnage de sols, zonage des parcelles, raisonnement par zones de gestion, dosages différenciés d'intrants, etc. Interpréter correctement les données massives collectées et prendre les bonnes décisions en fonction des analyses nécessite de nouvelles compétences pointues que les cursus de formation agricole traditionnels n'abordent pas ou peu. Le manque de main-d'œuvre qualifiée dans ce domaine peut constituer un frein majeur.

2. Les Applications de l'agriculture de précision

2.1 Gestion des sols et des cultures

Les scientifiques ont développé plusieurs systèmes de gestion des sols et des cultures (SGSC) pour promouvoir une utilisation optimale des ressources tout en préservant l'environnement. Ces SGSC visent à équilibrer les apports et les sorties de nutriments dans le sol en adaptant les quantités d'engrais aux besoins des cultures et en synchronisant leur application avec leur croissance. En utilisant des pratiques telles que l'agroforesterie, la rotation des cultures et l'irrigation, les SGSC améliorent la productivité des cultures tout en préservant les ressources en sol et en réduisant la pollution environnementale.

Cependant, l'utilisation accrue d'engrais azotés (N) et phosphatés (P) contribue à la pollution environnementale. En Chine, par exemple, les apports d'engrais ont augmenté

considérablement sans une augmentation proportionnelle des rendements agricoles, entraînant un déséquilibre des nutriments et une pollution accrue. Pourtant, l'amélioration de l'efficacité d'utilisation des engrais et l'adoption de pratiques de SGSC appropriées peuvent réduire ces risques environnementaux tout en augmentant les rendements agricoles.

En utilisant des engrais organiques et des pratiques de gestion du sol telles que le travail minimum, les agriculteurs peuvent améliorer la qualité du sol, séquestrer le carbone et réduire les émissions de gaz à effet de serre tout en augmentant les rendements en grains. En résumé, les SGSC offrent une approche prometteuse pour améliorer la productivité agricole tout en préservant l'environnement [7].

2.2 Détection et surveillance des ravageurs et des maladies des plantes

L'équipe éditoriale propose de concentrer nos efforts sur l'amélioration des technologies de surveillance et d'alerte précoce des principaux ravageurs et maladies en agriculture. Nous devrions notamment développer des technologies clés telles que l'identification intelligente des ravageurs et des maladies, le comptage automatique, et la reconnaissance d'images basée sur l'apprentissage automatique. Pour ce faire, nous devons investir dans la recherche et le développement de technologies de capture d'images au sol, de télédétection par drone à basse altitude, et de surveillance par radar à haute altitude. Il est également crucial d'établir une vaste base de données d'images de haute qualité. En parallèle, nous devrions travailler sur la collecte et le traitement des données massives, ainsi que sur le développement de technologies de surveillance et d'évaluation rapides grâce à l'Internet des objets et au cloud computing. En intégrant ces avancées technologiques, nous pourrons créer des modèles de prévision et d'alerte complets, basés sur une variété de sources d'informations, pour anticiper les cycles de reproduction des ravageurs et des maladies, les conditions de croissance des cultures, et les données météorologiques. Enfin, nous devons mettre en place une plateforme d'analyse intégrée, basée sur des cartes, permettant une surveillance en temps réel, une alerte précoce, et une diffusion efficace des informations sur les foyers émergents de ravageurs et de maladies [8].

2.3 Gestion de l'irrigation de précision

L'apprentissage automatique (ML) et l'apprentissage profond (DL) sont utilisés dans les systèmes d'irrigation intelligents pour la prédiction en temps réel et la prise de décision basée

sur les données historiques collectées par des capteurs et des systèmes IoT. ML et DL sont utilisés pour optimiser l'utilisation des ressources en eau, améliorer le rendement des cultures et maximiser les profits des agriculteurs. Ils sont également utilisés pour l'analyse de la protection des processus hydrologiques tels que l'humidité du sol et les niveaux d'eau souterraine. De plus, ML et DL sont utilisés pour optimiser le débit uniforme des émetteurs des systèmes d'irrigation goutte à goutte sous des conditions de pression et de température variables. Ces technologies jouent un rôle crucial dans l'irrigation intelligente basée sur la précision et dans le développement de modèles prédictifs pour des systèmes d'irrigation optimisés [9].

***Figure* I.2** irrigation intelligente [10].

3 Technologies de collecte de données

3.1. Systèmes de positionnement global (GPS)

Le système de positionnement par satellite NAVSTAR GPS, également connu sous le nom de Global Positioning System, est un système de navigation radio basé sur des satellites, développé et maintenu par le Département de la Défense des États-Unis. Il offre une

localisation précise en trois dimensions (latitude, longitude et élévation) à l'échelle mondiale, fonctionnant 24 heures sur 24, dans toutes les conditions météorologiques. Bien qu'à l'origine destiné à un usage militaire, le système est accessible aux civils avec certaines restrictions. Il est constitué d'un ensemble complet d'au moins 24 satellites en orbite autour de la Terre, disposés de manière stratégique pour assurer une couverture globale et une fiabilité maximale [11].

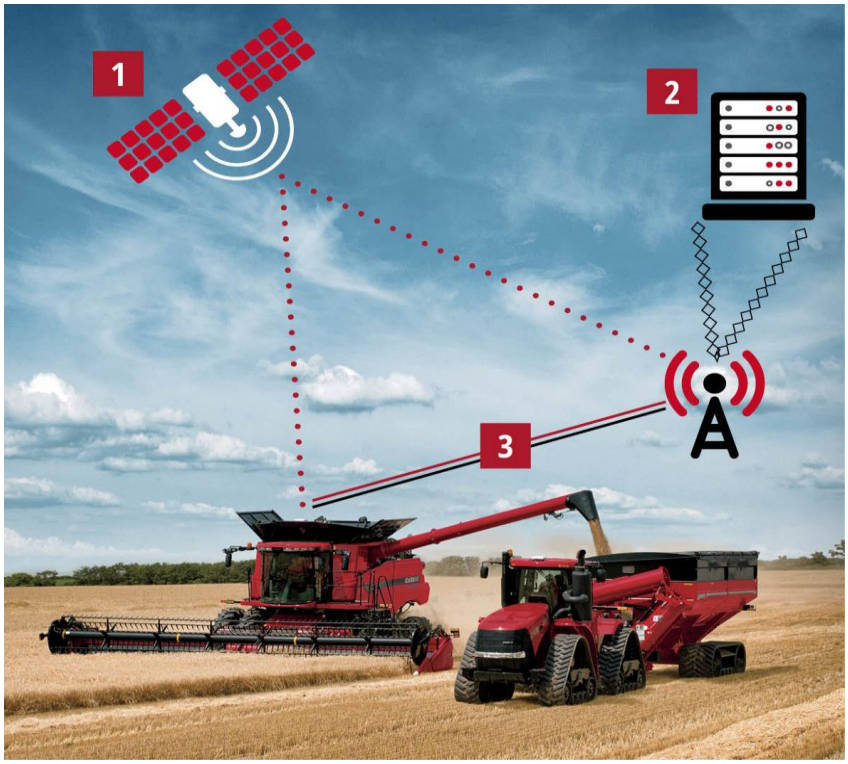


Figure I.3 Le fonctionnement du GPS dans l'agriculture de précision [12].

L'agriculture de précision consiste à optimiser les actions agricoles en fonction du bon moment, du bon endroit et de la bonne méthode. Cela implique l'utilisation de technologies avancées telles que le GPS pour collecter des données précises sur le terrain. Ces données sont ensuite intégrées dans des systèmes informatiques portables, permettant aux agriculteurs de visualiser et de modifier les informations en temps réel. Le système offre également la possibilité de sauvegarder les données GPS pour une utilisation ultérieure, ce qui facilite le suivi des activités sur le terrain. En outre, il permet d'améliorer la précision des analyses d'images satellites grâce à des données de vérité terrain précises et accessibles [13].

3.2 Drones agricoles

Un UAV (Véhicule Aérien Sans Pilote) est un appareil volant capable de suivre une trajectoire préétablie grâce à un pilote automatique et des coordonnées GPS. Il est également équipé de commandes radio classiques ; il peut être piloté manuellement en cas d'urgence ou de situation dangereuse. Parfois, le terme UAV est utilisé pour désigner l'ensemble du système, incluant les stations au sol et les systèmes vidéo, mais il est le plus souvent associé aux avions et hélicoptères modèles, qu'ils aient des ailes fixes ou rotatives [14].

Figure I.4 Drones dans l'agriculture de précision [15].

L'utilisation de drones dans l'agriculture moderne révolutionne plusieurs aspects de la gestion des terres cultivables. Tout d'abord, la surveillance de la santé des cultures en milieu de saison est grandement améliorée grâce à ces engins aériens équipés de capteurs sophistiqués comme le NDVI ou le NIR. Ces drones permettent une inspection efficace des cultures en croissance depuis une altitude d'environ 100 mètres, offrant une couverture étendue en moins de temps et réduisant les erreurs humaines associées aux méthodes traditionnelles d'inventaire. De même, la gestion de l'équipement d'irrigation devient moins contraignante grâce aux drones, en particulier pour les grands agriculteurs ayant de vastes étendues de terres. Ces drones facilitent notamment l'inspection des buses et des arroseurs sur plusieurs pivots d'irrigation, simplifiant ainsi la gestion de ces équipements, notamment lorsque les cultures comme le maïs atteignent une certaine hauteur en milieu de saison.

Par ailleurs, l'identification des mauvaises herbes au milieu des cultures est grandement facilitée par l'utilisation des données des capteurs NDVI. Les agriculteurs et leurs agronomes peuvent distinguer aisément les zones envahies de mauvaises herbes des zones

de cultures saines en traitant les images après le vol. Cette technologie permet une meilleure gestion des parcelles en permettant de prendre des mesures préventives contre les mauvaises herbes. De même, la fertilité à taux variable bénéficie également de l'utilisation des drones. En combinant les cartes NDVI avec des cartes d'application à taux variable, les agriculteurs peuvent optimiser l'application d'engrais en fonction des besoins spécifiques de chaque zone de leur champ, réduisant ainsi les coûts tout en augmentant les rendements.

Enfin, les drones jouent un rôle crucial dans la surveillance des troupeaux bovins, notamment pendant les périodes de prix bas des produits de base. Ils permettent de suivre la quantité et l'activité des animaux dans les champs, offrant ainsi une solution pratique pour surveiller les troupeaux, en particulier la nuit où la vision humaine est limitée. Cela aide les agriculteurs à détecter rapidement les problèmes et à prendre les mesures nécessaires pour assurer la santé et le bien-être de leurs troupeaux. En somme, l'utilisation des drones dans l'agriculture offre des avantages significatifs en termes de gestion des cultures, de l'irrigation, de la lutte contre les mauvaises herbes, de la fertilité des sols et de la surveillance des animaux, contribuant ainsi à une agriculture plus efficace et durable [16].

3.3 Systèmes d'information géographique (SIG)

Les Systèmes d'Information Géographique (SIG) permettent de visualiser des données traitées à partir d'images satellites ou aériennes, ainsi que des informations spatiales telles que des points, des polygones et des Shapefiles, via des bases de données géo-référencées. Aujourd'hui, une variété de SIG sont disponibles sur le marché, chacun se concentrant sur différents domaines tels que les urgences en cas de catastrophe, les changements dans l'utilisation des terres, l'urbanisme, l'énergie, la déforestation, et bien d'autres encore [17].

Figure I.5 Système d'Information Géographique [18].

Les SIG (Systèmes d'Information Géographique) sont composés de matériel, de logiciels, de données, d'individus et de méthodes, qui tous jouent un rôle essentiel dans leur fonctionnement. Le matériel comprend les ordinateurs sur lesquels les logiciels SIG sont exécutés, allant des serveurs informatiques centralisés aux ordinateurs de bureau en réseau. Les logiciels SIG fournissent les outils nécessaires pour stocker, analyser et afficher des informations géographiques, notamment des outils d'entrée et de manipulation de données, un système de gestion de base de données, des fonctionnalités d'analyse et de visualisation géographiques, ainsi qu'une interface utilisateur conviviale. Les données, qu'elles soient géographiques ou tabulaires, sont essentielles et peuvent être collectées en interne ou acquises auprès de fournisseurs commerciaux. Les individus jouent un rôle crucial dans la gestion et l'utilisation des SIG, allant des spécialistes techniques qui développent et maintiennent le système aux utilisateurs qui l'utilisent dans leurs tâches quotidiennes. Enfin, les méthodes et les règles métiers spécifiques à chaque organisation garantissent le bon fonctionnement et l'efficacité des SIG dans des contextes réels [19].

3.4 Les capteurs

Les capteurs, tels que les versions optiques de haute qualité, sont cruciaux pour l'agriculture, fournissant des données en temps réel sur la croissance des plantes, la fertilisation et la lutte contre les ravageurs. Ces informations sont essentielles pour les agriculteurs, notamment en Afrique, où elles permettent une utilisation plus efficace de l'eau et des nutriments. Les capteurs peuvent prendre diverses formes, y compris des smartphones simples, offrant aux

agriculteurs des conseils en ligne sur les traitements nécessaires. Les capteurs agricoles traditionnels, installés dans les champs ou sur des drones, permettent une agriculture plus intelligente et rentable en fournissant des données précises et spécifiques [20].

- **Capteurs de sol :** Ces capteurs mesurent l'humidité du sol, la teneur en nutriments et d'autres paramètres importants pour la croissance des cultures. Ils permettent aux agriculteurs de surveiller de près les conditions du sol et d'ajuster l'irrigation et la fertilisation en conséquence, ce qui peut augmenter les rendements tout en réduisant la consommation d'eau et de produits chimiques.
- **Capteurs météorologiques :** Ils surveillent les conditions météorologiques telles que la température, l'humidité, la vitesse du vent et les précipitations. Ces données peuvent aider les agriculteurs à prendre des décisions éclairées concernant la gestion des cultures, la planification des récoltes et la protection contre les conditions météorologiques extrêmes.
- **Capteurs de drones** : Les drones équipés de capteurs peuvent survoler les champs pour recueillir des données sur l'état des cultures, telles que la santé des plantes, la croissance, les infestations de ravageurs et les maladies. Cette surveillance permet aux agriculteurs d'identifier rapidement les problèmes potentiels et de prendre des mesures correctives précoces.
- **Capteurs de qualité de l'eau :** Dans les systèmes d'irrigation, les capteurs de qualité de l'eau surveillent la salinité et la qualité de l'eau utilisée pour l'irrigation. Cela aide à prévenir les dommages aux cultures causés par une irrigation excessive ou par une eau de mauvaise qualité.

Conclusion

L'agriculture de précision représente une véritable révolution dans les pratiques agricoles contemporaines. En tirant parti des technologies de pointe comme les drones, l'imagerie satellitaire, les systèmes de géolocalisation et les capteurs de sol, elle permet une gestion très fine et différenciée des intrants au niveau intra-parcellaire. Cette approche apporte de multiples bénéfices économiques, environnementaux et agronomiques, même si son déploiement à grande échelle reste confronté à certains défis.

Nul doute que l'agriculture de précision va continuer à se développer dans les années à venir, portée par les progrès technologiques constants dans des domaines comme le big data, l'intelligence artificielle ou encore la robotique agricole. Les solutions d'agriculture de précision de demain reposeront sur une intégration toujours plus poussée de ces briques technologiques innovantes.

C'est d'ailleurs l'un des sujets qui sera abordé dans le prochain chapitre, consacré à l'apprentissage automatique et son application au secteur agricole. Nous verrons comment les techniques d'apprentissage machine, en exploitant les masses de données collectées, peuvent permettre une prise de décision agro-environnementale optimisée et apporter une aide précieuse aux agriculteurs. Nous étudierons les fondements théoriques et méthodologiques de l'apprentissage automatique, avant d'illustrer son intérêt concret au travers de différents cas d'usage en agriculture.

CHAPITRE II
Les concepts de Machine Learning

Introduction

L'intelligence artificielle (IA) et le Machine Learning (ML) sont devenus des termes omniprésents dans notre monde technologique, mais leur signification et leurs implications restent souvent floues. Ce chapitre a pour objectif de démystifier ces concepts et de vous guider à travers les rouages de cette révolution technologique.

Nous commencerons par établir une base solide en définissant l'IA et le ML, en clarifiant leurs liens et leurs distinctions. Puis, nous plongerons dans le cœur du sujet : les différents types de Machine Learning. Vous découvrirez les approches supervisées, non supervisées et par renforcement, chacune ayant ses propres forces et applications.

Ensuite, nous explorerons les algorithmes qui animent le Machine Learning, des classiques comme la régression linéaire aux arbres de décision, en passant par les réseaux de neurones. Vous comprendrez comment ces algorithmes apprennent des données et effectuent des prédictions.

Le chapitre abordera également des concepts avancés tels que l'Ensemble Learning, une technique puissante qui combine plusieurs modèles pour améliorer les performances. Enfin, nous ouvrirons la porte sur le Deep Learning (DL) et les réseaux de neurones, une branche fascinante du ML inspirée par le fonctionnement du cerveau humain.

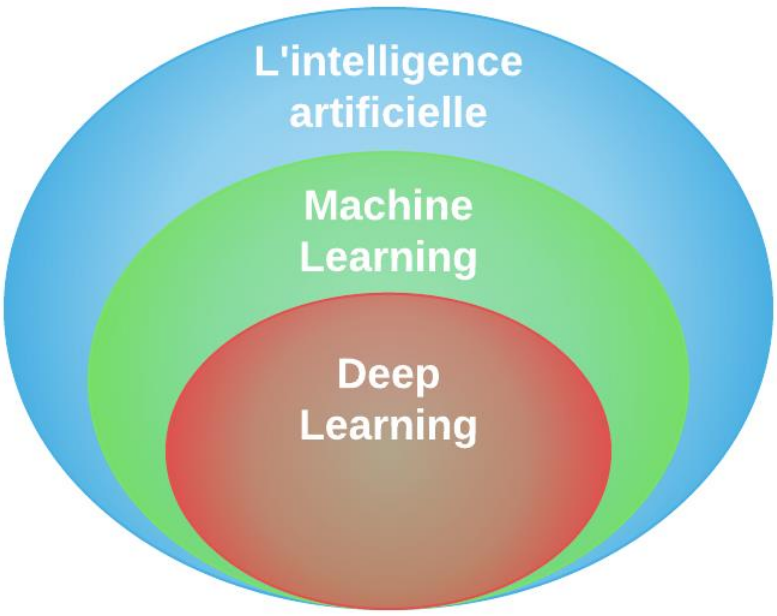


Figure *II.1 L'intelligence artificielle, Machine Learning et Deep Learning*

1 L’intelligence artificielle

L'intelligence artificielle, peut être définie comme "l'étude des agents intelligents : tout appareil qui perçoit son environnement et entreprend des actions qui maximisent ses chances de succès pour atteindre ses objectifs". En d'autres termes, l'IA concerne la création de systèmes capables de collecter des informations, d'apprendre, de s'adapter et de prendre des décisions rationnelles pour atteindre des objectifs définis. Cette définition met l'accent sur la rationalité et l'efficacité, soulignant la capacité des agents intelligents à optimiser leurs actions pour obtenir les meilleurs résultats possibles. [21]

2 L’apprentissage automatique (Machine Learning)

Le Machine Learning, ou apprentissage automatique, est un champ d'étude de l'intelligence artificielle qui se focalise sur la conception et le développement d'algorithmes permettant aux machines d'apprendre à partir de données et d'améliorer leurs performances sans être explicitement programmées. En d'autres termes, le ML permet aux ordinateurs de "s'entraîner" sur des jeux de données massifs et d'identifier des modèles et des relations qui leur permettent de faire des prédictions ou de prendre des décisions [22].

Le processus d'apprentissage automatique se déroule généralement en plusieurs étapes :

- **Collecte et préparation des données** : La qualité et la quantité des données sont cruciales pour la performance des algorithmes de ML.
- **Choix d'un modèle d’apprentissage :** Il existe une multitude de modèles de ML, chacun ayant ses propres forces et faiblesses. Le choix du modèle dépend de la nature du problème à résoudre et des données disponibles.
- **Entraînement du modèle :** Le modèle est nourri avec les données et ajuste ses paramètres internes pour minimiser l'erreur de prédiction.
- **Évaluation du modèle :** La performance du modèle est mesurée sur un jeu de données distinct pour s'assurer qu'il se généralise bien à de nouvelles données.
- **Déploiement et utilisation du modèle :** Si le modèle est performant, il peut être déployé et utilisé pour faire des prédictions ou prendre des décisions.

2.1 Types de machine Learning

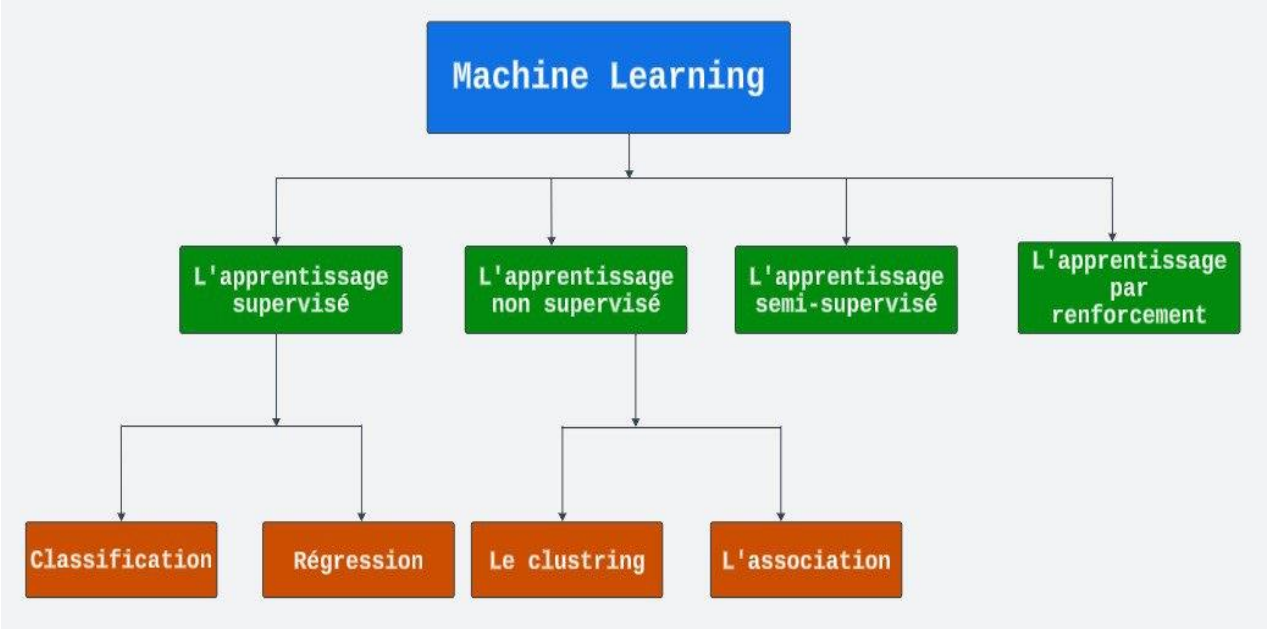


Figure II.2 Les types de Machine Learning

2.1.1 L'apprentissage supervisé

L'apprentissage supervisé est une approche en Machine Learning qui implique l'utilisation d'une vérité connue, c'est-à-dire que l'on dispose d'une connaissance préalable des valeurs de sortie attendues pour nos échantillons. Par conséquent, l'objectif de ce type d'apprentissage est de découvrir une fonction qui, à partir d'un ensemble de données d'entrée et des résultats désirés, se rapproche le plus possible de la relation entre les entrées et les sorties observables dans les données [23].

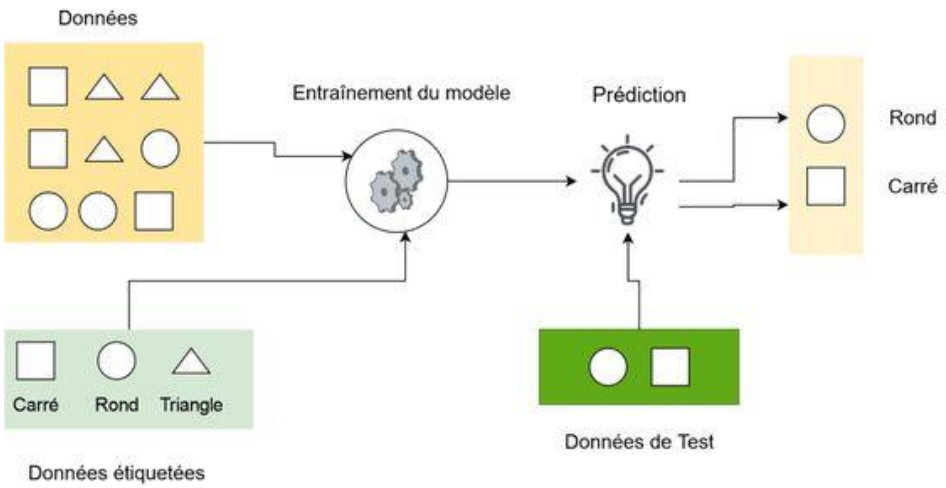


Figure II.3 l'apprentissage supervisé [24].

Dans le cadre de l'apprentissage supervisé, on distingue deux types d'algorithmes :

a. Classification

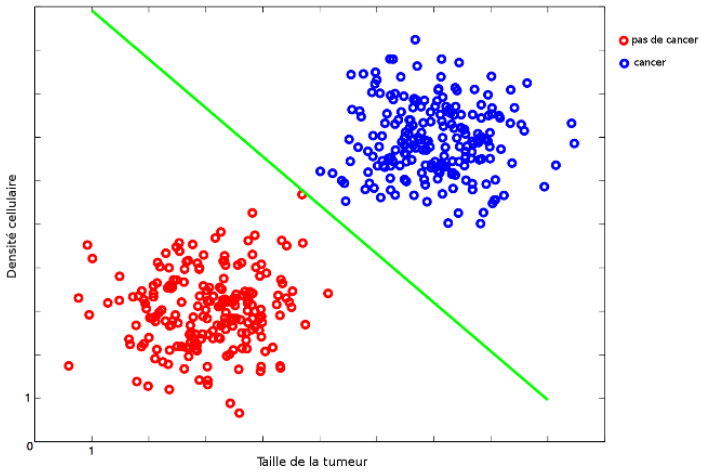


Figure II.4 Exemple de classification [25].

La classification est un processus qui consiste à assigner des étiquettes ou des catégories à des données en fonction de leurs caractéristiques. L'objectif est de prédire la classe d'une nouvelle observation en se basant sur les exemples d'entraînement. Par exemple, supposons que nous ayons un ensemble de données contenant des images d'animaux (chats, chiens et oiseaux). Un algorithme de classification pourrait apprendre à distinguer ces animaux en fonction des caractéristiques visuelles (par exemple, la forme des oreilles, la couleur du pelage, etc.). Lorsqu'une nouvelle image est présentée, l'algorithme peut prédire si elle représente un chat, un chien ou un oiseau. En résumé, la classification est un outil puissant pour résoudre divers problèmes, tels que la détection de spam, la reconnaissance d'objets, la prédiction de maladies et bien d'autres encore. Elle permet aux modèles d'apprentissage supervisé d'organiser et de catégoriser les données de manière efficace et précise.

Il existe trois types principaux de classification :

- **La classification binaire** est une tâche de Machine Learning dans laquelle le praticien doit classifier des données d'entrées entre deux catégories « classes ». Autrement dit, la classification binaire est le fait de diviser un ensemble de données « Dataset », en deux sous-ensembles distincts.
- **La classification multi-classe** est un type de problème de classification où les exemples sont assignés à l'une des trois classes ou plus. Contrairement à la classification binaire, qui ne comporte que deux classes, la classification multi-classe implique de distinguer entre plusieurs catégories possibles.

- **La classification multi-label** est un type d'apprentissage automatique où chaque exemple peut être associé à plusieurs catégories ou classes. Contrairement à la classification traditionnelle à étiquette unique, un exemple peut appartenir à plusieurs classes simultanément. Par exemple, une image peut être étiquetée à la fois comme "chat" et "animal domestique" [26].

  Il existe de nombreux types d'algorithmes de classification. Voici quelques-uns parmi les plus courants :

- **L'Algorithme K-Nearest Neighbors (KNN)**

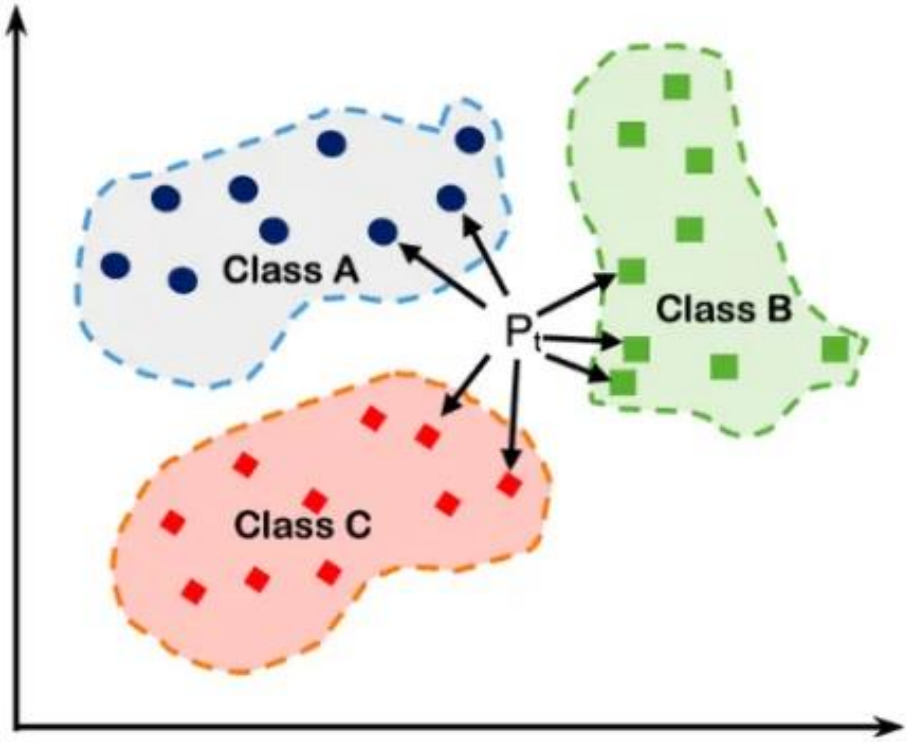


Figure II.5 Exemple de l'algorithme KNN [27].

Est une méthode de classification qui s'appuie sur la similarité entre les points de données. Pour classer un nouveau point, KNN identifie les K points de données les plus proches (ses "voisins") dans l'ensemble d'apprentissage. La classe majoritaire parmi ces K voisins est ensuite attribuée au nouveau point. Le choix du nombre de voisins (K) est crucial et influence la performance de l'algorithme.

- **La classification SVM (Support Vector Machine)** est une technique d'apprentissage automatique supervisé qui utilise des machines à vecteurs de support (SVM) pour classer des données. Elle fonctionne en trouvant le meilleur hyperplan qui sépare les différentes classes de données dans un espace multidimensionnel, en maximisant la marge entre l'hyperplan et les points de données les plus proches de chaque classe (vecteurs de support) .

- **Un arbre de décision** est un algorithme de Machine Learning supervisé qui utilise une structure arborescente pour classer les données. Chaque nœud de l'arbre représente une question sur une caractéristique, et chaque branche représente une réponse possible. Les feuilles de l'arbre correspondent aux classes possibles [28].

- **La régression logistique** est un algorithme d'apprentissage automatique supervisé utilisé pour prédire la probabilité qu'une observation appartienne à une catégorie binaire (par exemple, oui/non, vrai/faux). Il utilise une fonction logistique pour modéliser la relation entre les variables indépendantes et la variable dépendante binaire [29].

- **Une forêt aléatoire (Random Forest)** est un algorithme de classification qui combine plusieurs arbres de décision. Chaque arbre est entraîné sur un sous-ensemble aléatoire des données d'apprentissage et des variables d'entrée. Les forêts aléatoires sont robustes au bruit et aux valeurs aberrantes et peuvent gérer des données avec un grand nombre de variables d'entrée [30].

b. Régression

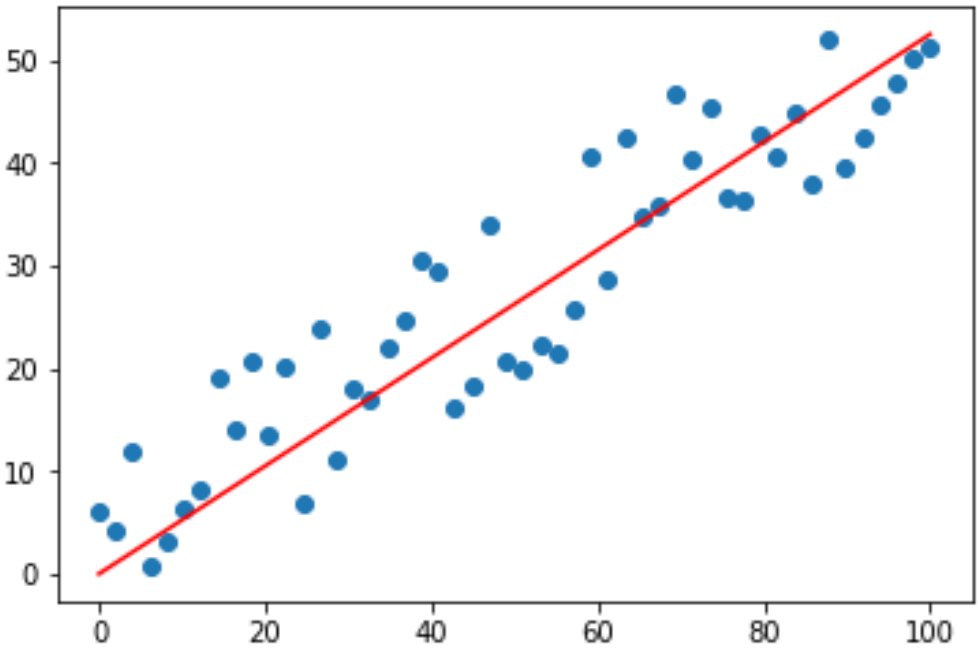


Figure II.6 Exemple de régression [31].

En apprentissage automatique, la régression est une technique qui permet de modéliser et de prédire des valeurs continues à partir de données d'entraînement. Elle est utilisée pour estimer la relation entre une variable dépendante et une ou plusieurs variables indépendantes. Par exemple, estimer le prix d'une maison en fonction de sa superficie, son nombre de chambres et son emplacement. Les deux algorithmes de régression les plus connus semblent être :

Régression linéaire : La régression linéaire cherche à modéliser une relation linéaire entre une variable à prédire (la sortie) et une ou plusieurs variables explicatives (les entrées). Elle estime les paramètres (poids) d'une équation linéaire à partir de données d'entraînement.

Régression polynomiale : La régression polynomiale est une extension de la régression linéaire où de nouvelles caractéristiques sont créées en élevant les variables d'entrée à différentes puissances polynomiales. Cela permet de capturer des relations non linéaires entre les entrées et la sortie [32].

2.1.2 L'apprentissage non supervisé

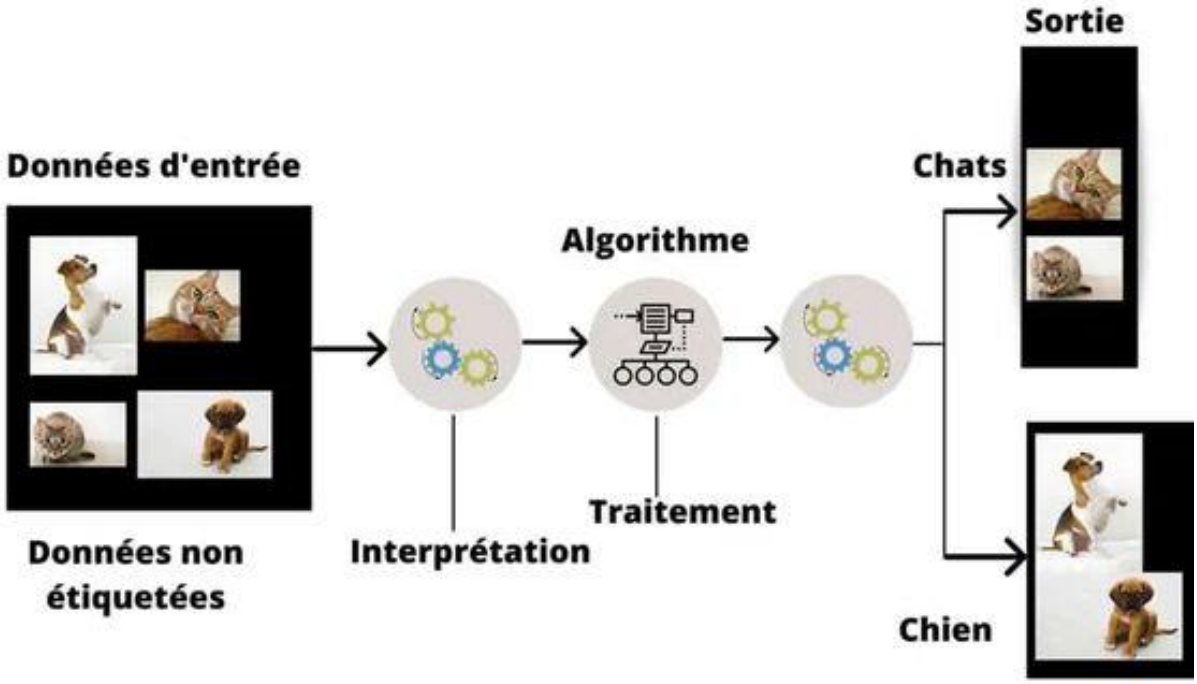


Figure II.7 l'apprentissage non supervisé [23].

L'apprentissage non supervisé représente un pan de l'apprentissage automatique dans lequel les algorithmes analysent et regroupent des données dépourvues d'étiquettes préalables. Ces méthodes découvrent des structures, motifs ou catégories intrinsèques présents au sein des données, et ce, avec un minimum d'intervention humaine. Cette approche diffère de l'apprentissage supervisé, où le modèle est alimenté avec des données d'entraînement labellisées pour apprendre. En effet, les modèles supervisés et non supervisés se distinguent par le type de données qu'ils utilisent. Dans l'apprentissage supervisé, les données d'entrée et de sortie sont étiquetées, tandis que dans l'apprentissage non supervisé, le modèle apprend à partir de données d'entraînement non étiquetées pour effectuer des prédictions de classification. Ainsi, l'objectif d'un modèle non supervisé est de découvrir des structures ou des schémas dans un ensemble de données volumineux, alors que dans un modèle supervisé, l'objectif est de prédire les valeurs de sortie pour de nouvelles données.

L'apprentissage non supervisé peut être divisé en deux grands types de techniques : les méthodes de Clustering (ou de regroupement) et les méthodes d'association (ou de règles d'association).

a. **Le Clustering (regroupement)** : est un processus où la machine regroupe des objets dans des ensembles distincts, appelés clusters, en fonction de leurs similarités dans un ensemble

de données. L'objectif est de créer des groupes de manière juste et efficace. Bien que parfois complexe pour les humains à saisir, cette technique est largement utilisée dans des domaines tels que le marketing pour segmenter les clients en fonction de leurs comportements ou préférences. Un algorithme couramment employé pour le Clustering est le K-means.

b. L'association de données consiste à identifier et à regrouper des éléments ayant des caractéristiques similaires, même s'ils ne sont pas nécessairement identiques. Par exemple, en fournissant à un algorithme un ensemble d'images comprenant des chats et des accessoires pour chats, le système pourrait regrouper une pelote de laine avec un chat, plutôt que de simplement regrouper tous les chats ensemble [33]. L'objectif est de découvrir des relations entre les données en se basant sur des similitudes spécifiques. Un exemple courant d'algorithme utilisé dans l'association est l'algorithme A-priori.

2.1.3 L'apprentissage semi-supervisé

Est une technique d'apprentissage automatique qui se situe entre l'apprentissage supervisé et l'apprentissage non supervisé. Il utilise un ensemble de données composé à la fois de données étiquetées (avec des réponses connues) et de données non étiquetées (sans réponses connues) pour entraîner un modèle [34]. L'objectif est d'exploiter les informations des données étiquetées pour mieux comprendre la structure des données non étiquetées et ainsi améliorer la performance du modèle sur des tâches spécifiques, telles que la classification ou la régression. Par exemple : Entraîner un modèle pour classer des images d'animaux (éléphants, chiens, serpents) en utilisant un petit nombre d'images étiquetées et un grand nombre d'images non étiquetées. L'apprentissage semi-supervisé apprend des images étiquetées et applique ensuite cette connaissance pour classer les images non étiquetées.

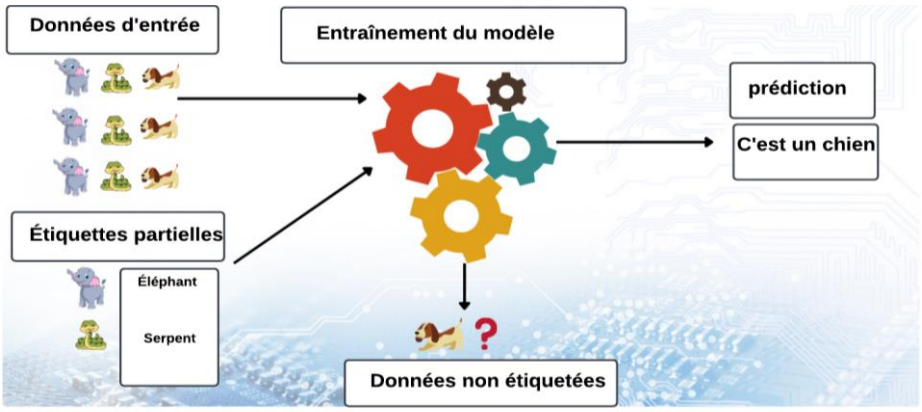


Figure II.8 l'apprentissage semi- supervisé

2.1.4 L'apprentissage par renforcement

Il s'agit d'une méthode de Machine Learning où un agent apprend à prendre des décisions en interagissant avec un environnement. L'agent reçoit des récompenses pour les bonnes actions et des pénalités pour les mauvaises actions, et il ajuste progressivement sa stratégie pour maximiser les récompenses cumulées à long terme. Ce processus s'apparente à la façon dont les humains apprennent par essais et erreurs, en tirant des leçons de leurs expériences passées [35]. Par exemple Un robot apprend à naviguer dans un labyrinthe. Il reçoit une récompense lorsqu'il atteint la sortie et une pénalité lorsqu'il heurte un mur. Au fil du temps, le robot apprend à éviter les murs et à trouver le chemin le plus rapide vers la sortie.

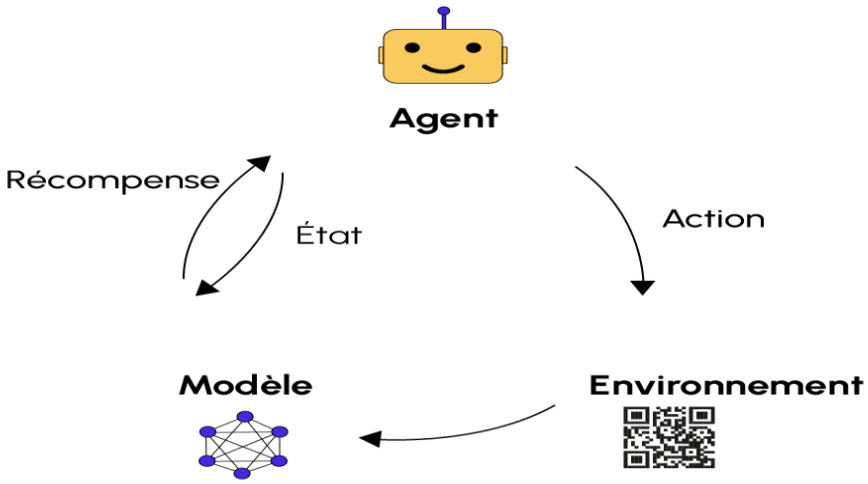


Figure II.9 L'apprentissage par renforcement [36].

3 L'apprentissage Profond (Deep Learning)

L'apprentissage profond, est un sous-domaine de l'apprentissage automatique qui s'appuie sur des réseaux de neurones artificiels comportant de multiples couches. Ces réseaux sont inspirés du fonctionnement du cerveau humain et sont capables d'apprendre des représentations complexes des données à partir d'un grand volume d'exemples. Le Deep Learning excelle dans des tâches telles que la reconnaissance d'images, la compréhension du langage naturel et la traduction automatique, où les données sont souvent non structurées et complexes [37].

Le Deep Learning a connu un succès considérable dans divers domaines, notamment :

- **Reconnaissance d'images et de vidéos** : Identification d'objets, de scènes, de visages, etc.
- **Traitement du langage naturel** : Traduction automatique, analyse de sentiments, génération de texte.
- **Reconnaissance vocale** : Transcription de la parole en texte, assistants vocaux.

- **Jeux :** Les programmes de Deep Learning ont battu les champions du monde dans des jeux complexes comme Go et les échecs.
- **Agriculture :** Le Deep Learning est de plus en plus utilisé dans l'agriculture pour des tâches telles que la détection des maladies des plantes, la surveillance des cultures, l'analyse des sols et la gestion de l'irrigation.

3.1 Les réseaux de neurones artificiels (ANN)

Également appelés réseaux neuronaux, sont des modèles informatiques inspirés du fonctionnement du cerveau humain. Ils sont composés d'unités interconnectées appelées neurones artificiels, qui traitent et transmettent des informations. Chaque neurone reçoit des signaux en entrée, les pondère, et produit un signal de sortie qui peut être transmis à d'autres neurones. L'apprentissage dans un réseau de neurones consiste à ajuster les poids des connexions entre les neurones afin d'améliorer la performance du réseau sur une tâche spécifique [38].

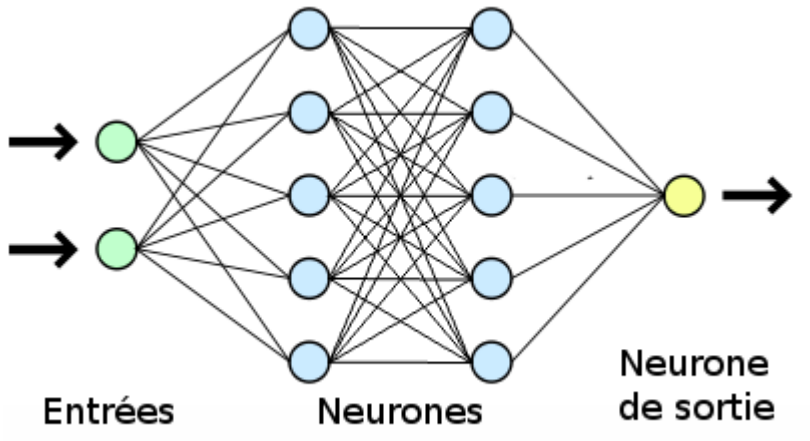


Figure II.10 Architecture d'un réseau de neurones artificiels.

4 Stratégies Avancées d'Apprentissage

Les stratégies avancées d'apprentissage sont des techniques visant à étendre et optimiser les capacités des modèles d'apprentissage automatique traditionnels. Elles permettent de relever certains défis majeurs tels que la pénurie de données étiquetées, le transfert de connaissances entre domaines connexes ou la combinaison de multiples modèles. Ces stratégies exploitent des concepts comme le pré-entraînement sur de grandes quantités de données non étiquetées, l'adaptation à de nouvelles tâches, l'apprentissage décentralisé ou encore la fusion de plusieurs modèles. Elles offrent ainsi des solutions pour tirer le meilleur parti des données disponibles, réutiliser efficacement les connaissances existantes et combiner intelligemment différentes

approches. Parmi les principales stratégies avancées d'apprentissage, on peut citer le Fine-Tuning, l'apprentissage fédéré, le transfert d'apprentissage et l'ensemble d'apprentissage.

4.1 Le Fine-Tuning

Le Fine-Tuning, ou "affinage", est une technique d'apprentissage par transfert utilisée en intelligence artificielle, en particulier dans le domaine de l'apprentissage profond. Il s'agit d'un processus de réentraînement d'un modèle pré-entraîné sur une nouvelle tâche ou un nouveau domaine de données, en utilisant un plus petit ensemble de données spécifiques à cette tâche. Les modèles d'IA modernes, comme les grands modèles de langage (GPT-3, BERT, etc.), sont initialement entraînés sur d'énormes quantités de données générales, ce qui leur permet d'acquérir une compréhension générale du langage et du monde. Cependant, ces modèles peuvent manquer de performances sur des tâches ou des domaines spécifiques. C'est là que le Fine-Tuning intervient. Le processus de Fine-Tuning consiste à prendre un modèle pré-entraîné et à ajuster ses paramètres sur un plus petit ensemble de données spécifiques à la nouvelle tâche. Cela permet au modèle de transférer les connaissances acquises lors de son pré-entraînement initial et de les adapter à la nouvelle tâche, plutôt que de devoir être entièrement ré-entraîné à partir de zéro.

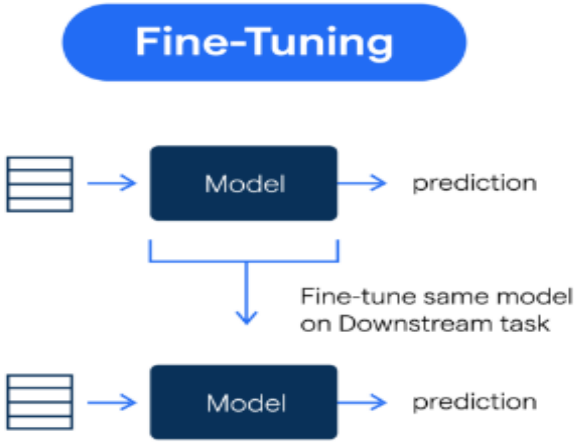


Figure II.11 Processus de Fine-Tuning [39].

4.1.1 Les avantages du Fine-Tuning

. Performances accrues sur des tâches spécifiques : En adaptant un modèle pré-entraîné à une tâche particulière, le Fine-Tuning permet d'améliorer les performances du modèle sur cette tâche, comparé à un modèle généraliste non affiné.

. Économie de ressources : Le Fine-Tuning nécessite généralement beaucoup moins de données et de ressources de calcul que l'entraînement complet d'un modèle depuis le début, ce qui le rend plus économique et plus rapide.

. Transfert de connaissances : En partant d'un modèle pré-entraîné, le Fine-Tuning permet de transférer et de réutiliser les connaissances acquises lors du pré-entraînement initial, plutôt que de tout réapprendre à partir de zéro.

. Adaptation à des domaines spécifiques : Le Fine-Tuning permet d'adapter un modèle généraliste à des domaines ou des tâches spécifiques, comme la reconnaissance d'entités nommées dans un domaine particulier, la traduction dans un domaine technique, etc.

Le Fine-Tuning est largement utilisé dans diverses applications de l'IA, comme le traitement du langage naturel, la vision par ordinateur, la reconnaissance de la parole, etc. C'est une technique cruciale pour tirer parti des grands modèles pré-entraînés et les adapter à des tâches spécifiques de manière efficace.

4.1.2 Les défis liés au Fine-Tuning

Un des défis majeurs du Fine-Tuning est le risque de Catastrophic Forgetting (oubli catastrophique). Comme les modèles sont ré-entraînés sur de nouveaux ensembles de données spécifiques, ils peuvent "oublier" une partie des connaissances générales apprises lors du pré-entraînement initial. Il faut donc trouver un équilibre pour conserver les connaissances préalables tout en adaptant le modèle à la nouvelle tâche.

Un autre défi est la taille limitée des ensembles de données disponibles pour le Fine-Tuning dans de nombreux domaines spécialisés. Avec trop peu de données, le modèle affiné risque de ne pas bien généraliser et de souffrir de sur-apprentissage (Overfitting). Il faut donc parfois constituer manuellement ces données, ce qui peut être coûteux.

De plus, le Fine-Tuning nécessite de choisir des hyper-paramètres optimaux comme le taux d'apprentissage, le nombre d'époques, etc. Ce réglage fin est crucial pour obtenir de bonnes performances, mais peut s'avérer complexe et chronophage.

Enfin, dans certains cas, le Fine-Tuning peut introduire des biais indésirables dans le modèle s'il est effectué sur des données non représentatives ou biaisées. Il faut donc veiller à la qualité et à la diversité des données utilisées [40].

4.2 Transfer Learning

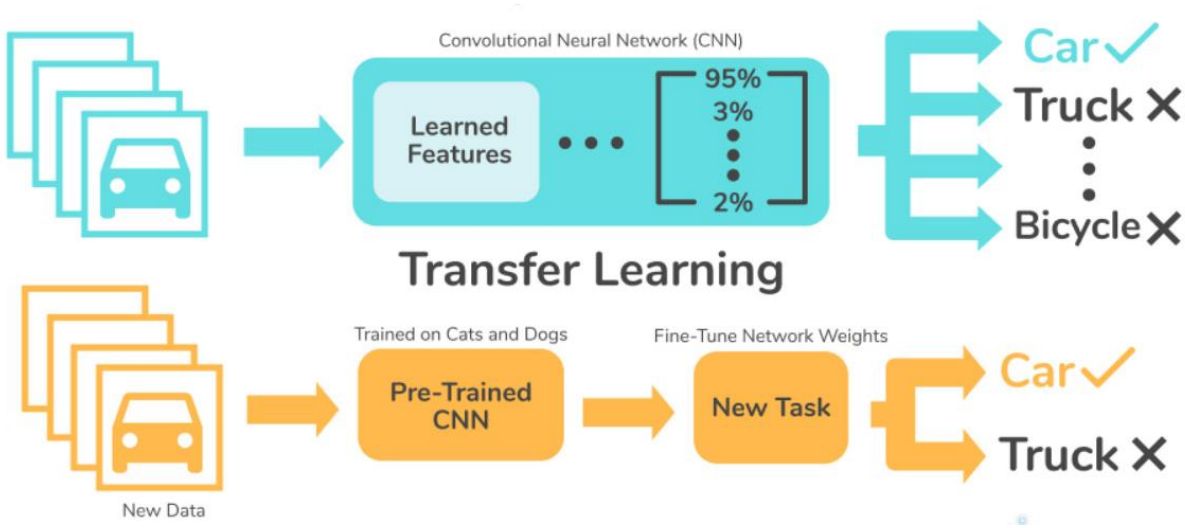


Figure II.12 les étapes clés du processus de Transfer Learning [41].

Le Transfer Learning est une méthode d'apprentissage automatique qui consiste à exploiter les connaissances acquises lors d'une tâche source pour améliorer l'apprentissage sur une nouvelle tâche cible connexe. Au lieu d'entraîner un modèle à partir de zéro, on réutilise ce qui a été appris sur une tâche similaire. Cela permet d'obtenir de meilleures performances prédictives sur la tâche cible, tout en nécessitant beaucoup moins de données d'entraînement. Un exemple courant est l'utilisation de réseaux de neurones convolutifs pré-entraînés pour des tâches de reconnaissance d'images avec peu de données annotées. il existe trois grandes catégories de Transfer Learning :

Induction par transfert : Cette catégorie vise à réutiliser les connaissances extraites des tâches sources pour construire un modèle de classification ou de régression adaptée à la tâche cible. Les connaissances transférées peuvent prendre diverses formes comme des instances, des règles d'apprentissage ou des caractéristiques.

Transfert de représentation (Feature-Representation Transfer) : L'objectif est ici de représenter les données des tâches sources et cibles dans un même espace de caractéristiques pour réduire la divergence entre leurs distributions. On cherche des représentations permettant de minimiser les différences entre domaines tout en préservant les propriétés discriminantes.

Échantillonnage d'instances : Cette approche consiste à ré-échantillonner les instances sources, c'est-à-dire ajuster leur poids/importance, afin que leur distribution soit mieux alignée avec la tâche cible. Cela permet de transférer plus efficacement la connaissance des instances sources .

4.2.1 Les avantages du Transfer Learning

Gain de temps et d'efficacité : Le Transfer Learning permet de réutiliser les connaissances acquises par un modèle pré-entraîné sur une grande quantité de données, évitant ainsi d'avoir à entraîner un nouveau modèle depuis zéro.

Besoin de moins de données : Grâce au transfert des connaissances du modèle pré-entraîné, on peut obtenir de bonnes performances même avec une quantité de données d'entraînement limitée pour la tâche finale.

Performances accrues : En initialisant avec des poids pré-entraînés, on part d'un meilleur point de départ qu'avec une initialisation aléatoire, ce qui peut conduire à de meilleures performances finales.

Régularisation : Le Transfer Learning peut être vu comme une forme de régularisation qui aide à éviter le sur-apprentissage.

4.2.2 Les principaux défis liés au Transfer Learning

Divergence entre les domaines source et cible : Lorsque les données de la tâche source et de la tâche cible proviennent de domaines très différents, le transfert des connaissances peut être difficile ou inefficace.

Sélection appropriée du modèle pré-entraîné : Il est important de choisir un modèle pré-entraîné approprié pour la nouvelle tâche, en termes de capacité, architecture et domaine d'application.

Réglage fin du modèle transféré : Trouver le bon équilibre entre figer certaines parties du réseau et entraîner le reste, afin de bien transférer les connaissances utiles sans "oublier" ce qui a été appris.

Temps de calcul : Bien que plus rapide qu'un entraînement complet, le Transfer learning peut nécessiter des temps de calcul importants selon la complexité des modèles.

Manque de données annotées pour le pré-entraînement : Pour certains domaines, il peut être difficile de disposer de grandes quantités de données annotées pour pré-entraîner efficacement les modèles [42].

4.3 Federated Learning

L'apprentissage fédéré est une technique collaborative de Machine Learning dans laquelle plusieurs clients, comme des smartphones ou des dispositifs en périphérie, entraînent un modèle sur leurs propres données locales. Les mises à jour locales sont ensuite agrégées par un serveur central pour former un modèle global. Ce processus itératif permet d'améliorer le modèle global tout en préservant la confidentialité des données [43].

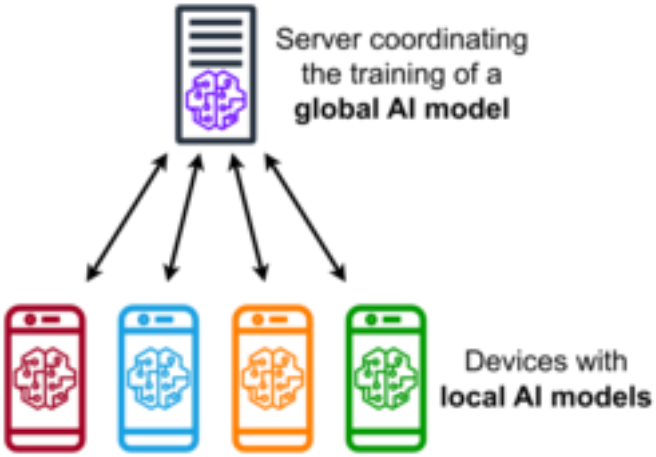


Figure II.13 Le fonctionnement de L'apprentissage fédéré [44].

4.3.1 Le fonctionnement de L'apprentissage fédéré

Dans un système de Federated Learning, chaque appareil participant au réseau possède sa propre copie du modèle. Ces appareils entraînent ensuite leur propre modèle en utilisant leurs données locales. Les paramètres ou pondérations des modèles individuels sont ensuite transmis à un appareil central, souvent appelé serveur, qui agrège ces paramètres et met à jour le modèle global. Ce processus itératif peut se répéter jusqu'à ce que le niveau de précision désiré soit atteint.

4.3.2 Les avantages de L'apprentissage fédéré

- Confidentialité des données : Les données brutes restent sur l'appareil du client, diminuant ainsi les risques pour la vie privée.

- Réduction de la bande passante : Seules les mises à jour du modèle sont envoyées, et non les ensembles de données entiers.

- Personnalisation : Les modèles peuvent être adaptés selon les données locales, ce qui améliore les performances pour chaque utilisateur individuel.

4.3.3 Les Limitations de l'Apprentissage Fédéré

L'apprentissage fédéré permet à plusieurs collaborateurs de former des modèles sur leurs données locales et d'agréger ces modèles pour créer un modèle généralisé. Idéalement, tous les modèles locaux apprennent des patterns similaires, facilitant leur agrégation. Cependant, des différences entre les populations de données des collaborateurs peuvent entraîner des

modèles divergents, ralentissant la convergence. Cette lenteur augmente les coûts d'infrastructure et de réseau, et ralentit les cycles d'itération et d'expérimentation, rendant plus difficile le déploiement rapide des meilleurs modèles en production.

4.3.4 Les défis de l'Apprentissage Fédéré

- **Communication Coûteuse** : Dans les réseaux fédérés, qui peuvent comporter de nombreux appareils tels que des smartphones, la communication peut être beaucoup plus lente que les calculs locaux, ce qui entraîne des coûts de communication élevée. Pour adapter un modèle aux données des appareils dans un réseau fédéré, il est essentiel de développer des méthodes de communication efficaces qui envoient des messages ou des mises à jour de modèle de manière itérative pendant l'entraînement, plutôt que de transférer l'ensemble des données sur le réseau.
- **Hétérogénéité des Systèmes** : Les réseaux fédérés présentent des disparités importantes en termes de capacités matérielles, de connectivité réseau et de niveau de batterie entre les appareils individuels. De plus, dans un réseau composé de millions d'appareils, seules quelques centaines sont souvent actives simultanément, et la fiabilité des appareils peut être sujette à des problèmes, certains pouvant cesser de fonctionner à tout moment. Ces caractéristiques systémiques rendent les problèmes tels que les traînards et la tolérance aux pannes bien plus fréquents dans les réseaux fédérés que dans les environnements de centre de données traditionnels.
- **Hétérogénéité Statistique** : Les dispositifs au sein des réseaux fédérés génèrent et rassemblent souvent des données de manière inégale. Par exemple, les utilisateurs de téléphones mobiles peuvent avoir des schémas d'utilisation linguistique très divers. De plus, la quantité de données échangées entre les dispositifs peut varier considérablement, et il peut exister une structure sous-jacente qui décrit les relations entre les dispositifs et leurs distributions de données. Cette diversité statistique rend les hypothèses traditionnelles d'une distribution identique et indépendante (I.I.D) moins applicables en optimisation distribuée, ce qui peut complexifier l'analyse des données et des modèles.
- **Préoccupations Concernant la Confidentialité** : La préservation de la confidentialité dans les réseaux fédérés est cruciale, car les mises à jour de modèle transmises peuvent inadvertamment divulguer des informations sensibles, même si les données originales demeurent sur les appareils locaux. Bien que des approches telles que la confidentialité différentielle aient été proposées pour améliorer la sécurité, leur utilisation peut

entraîner une diminution des performances ou de l'efficacité du système. Ainsi, équilibrer la confidentialité et la performance demeure un défi de taille dans le déploiement de systèmes d'apprentissage fédéré sécurisés [45].

4.4 Ensemble Learning

L'ensemble Learning fait référence à un ensemble de techniques d'apprentissage automatique qui construisent un prédicteur en combinant les forces de plusieurs modèles individuels. Au lieu d'utiliser un seul modèle d'apprentissage, l'idée est d'en construire une multitude, en exploitant leur diversité et en les combinant de manière intelligente. L'idée de base de l'ensemble Learning est d'apprendre non pas un seul prédictif mais un ensemble de prédicteurs, chacun avec une certaine capacité de prédiction, et de les combiner de manière à obtenir un prédicteur d'ensemble plus précis [46]. Parmi les techniques couramment utilisées, on retrouve le Bagging, le Boosting et le Stacking.

4.4.1 Bagging

Le Bagging (Bootstrap Aggregating) est une méthode d'ensemble qui vise à réduire la variance d'un estimateur de Machine Learning. L'idée est d'entraîner plusieurs modèles de base sur des sous-ensembles de données bootstrappés puis de les combiner par vote majoritaire ou moyenne. Le Bagging manipule efficacement la variance instable du processus d'apprentissage en introduisant de façon aléatoire une variabilité supplémentaire dans le constructeur. Bien que les prédictions des modèles individuels soient potentiellement erronées, la moyenne de leurs prédictions dérivée du Bagging peut donner une prédiction agrégée très précise [47]. Un célèbre algorithme de Bagging est le fameux Random Forest.

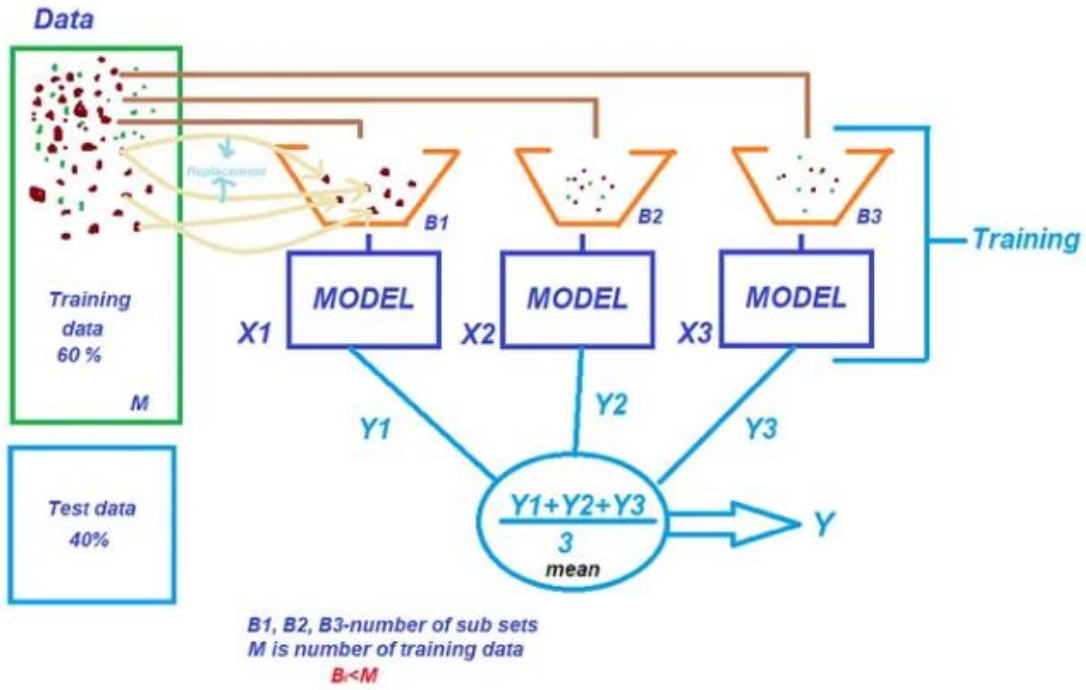


Figure II.14 Schéma du processus de Bagging (Bootstrap Aggregating) [48].

4.4.2 Boosting

Le Boosting fait référence à une famille d'algorithmes qui combinent les sorties de plusieurs modèles faibles et imparfaits pour produire un prédicteur puissant. À chaque itération, un nouveau modèle est entraîné sur les erreurs du modèle précédent, de sorte que les modèles successifs se concentrent sur les exemples difficiles. L'algorithme de Boosting produit un ensemble de prédicteurs faibles. Les exemples qui sont mal prédits obtiennent un poids de priorité plus élevé pour être bien représentés dans la formation des prédicteurs faibles ultérieurs. Ainsi, le Boosting alterne entre les étapes d'ajustement des pondérations et d'ajustement des prédicteurs faibles, de sorte que les exemples mal prédits obtiennent un poids de plus en plus élevé à chaque itération [49].

Les algorithmes de Boosting couramment utilisés :

- **Gradient Boosting (GBM) :** GBM construit le modèle de manière séquentielle, comme les autres méthodes de Boosting, et il permet d'optimiser une fonction de perte arbitraire. À chaque étape, une fonction de perte est modélisée avec une fonction simple, comme une décision d'arbre [50].

- **XGBoost :** XGBoost est une implémentation très efficace et parallélisable de l'algorithme de gradient Boosting pour arbres de décision. Il tire parti des systèmes de calcul parallèle et distribué pour améliorer considérablement les performances de calcul [51].
- **LightGBM :** LightGBM est une autre implémentation populaire de l'algorithme de gradient Boosting proposée par Microsoft. Il est conçu pour être très efficace, avec des temps d'entraînement encore plus rapides que XGBoost et une utilisation économe des ressources système, tout en conservant une précision élevée [52].

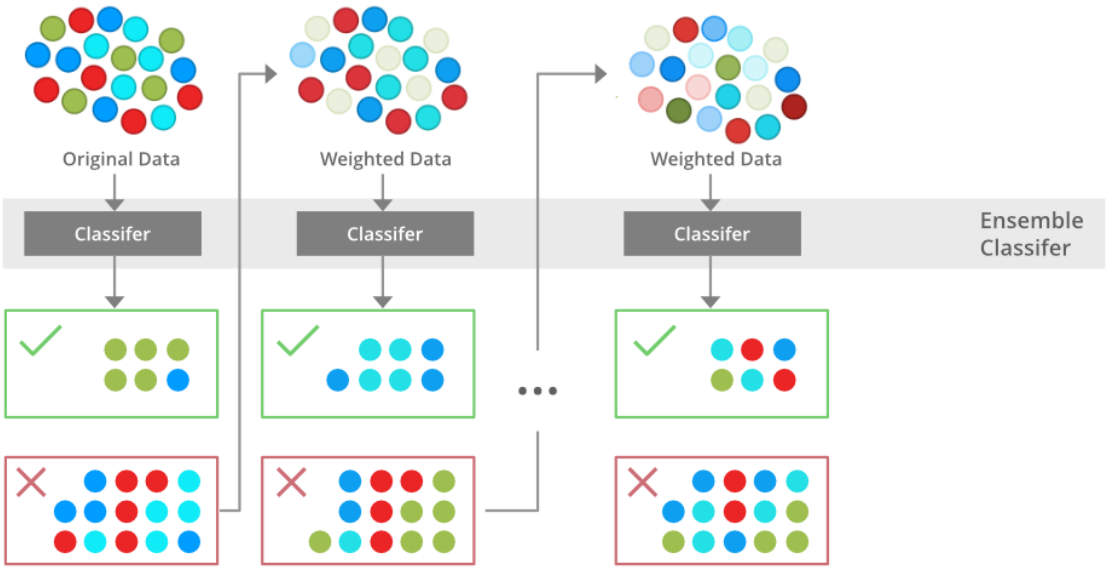


Figure II.15 Entraînement d'un modèle de Boosting [53].

4.4.3 Stacking

Le Stacking, ou Stacked Generalization, implique d'entraîner un modèle d'apprentissage, appelé méta-modèle, pour combiner les prédictions de plusieurs modèles de base. Le méta-modèle apprend comment assigner des poids optimaux aux différents modèles de base. Dans la méthode de Stacking, nous combinons les sorties de plusieurs modèles d'apprentissage à l'aide d'un méta-modèle d'apprentissage. Le méta-modèle est formé pour combiner de manière optimale les sorties des modèles de base en fonction de leurs performances respectives sur les données d'entraînement [54].

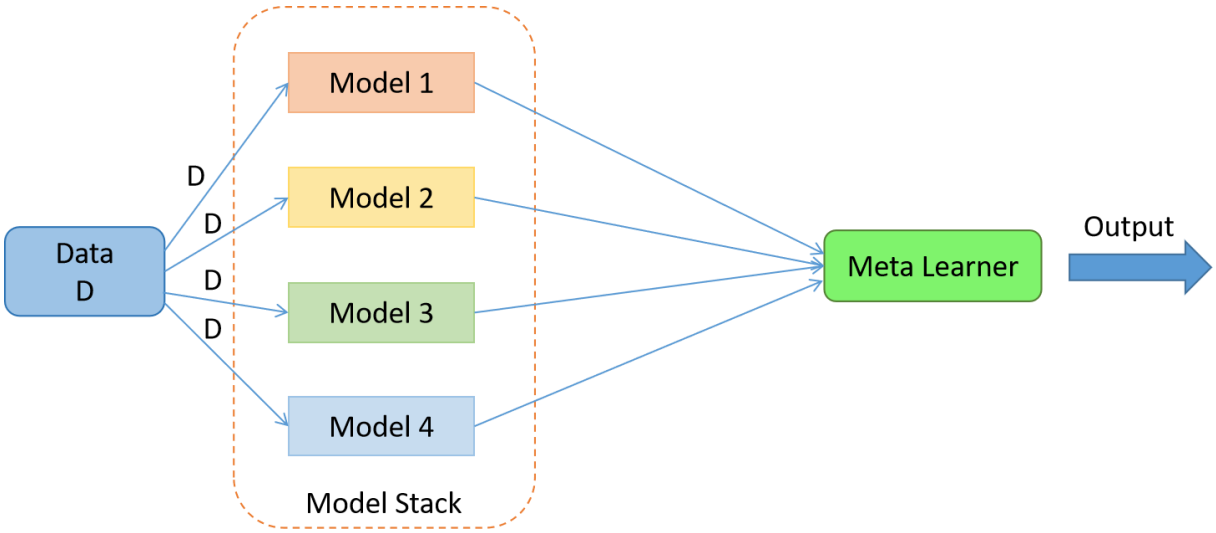


Figure II.16 Présentation du processus d'apprentissage avec Stacking [55].

Conclusion

Dans ce chapitre, nous avons exploré les concepts fondamentaux de l'intelligence artificielle (IA) et du Machine Learning. Nous avons vu que le Machine Learning permet aux systèmes d'apprendre et de s'améliorer à partir de données, sans être explicitement programmés. Le Deep Learning, une branche du machine Learning inspirée du fonctionnement des neurones biologiques, s'est imposé comme une approche particulièrement puissante pour traiter des données complexes comme les images, les textes ou les signaux.

Nous avons également abordé des stratégies avancées d'apprentissage comme le Fine-Tuning, qui consiste à adapter un modèle pré-entraîné à une nouvelle tâche, le Transfer Learning qui réutilise les connaissances d'un modèle sur un nouveau problème, et le Federated Learning qui permet d'entraîner des modèles de manière décentralisée sur des données distribuées. L'ensemble Learning, qui combine plusieurs modèles, a aussi été présenté.

Ces techniques offrent de nouvelles perspectives passionnantes pour relever les défis de l'agriculture de précision, sujet qui sera abordé dans le prochain chapitre. L'exploitation intelligente des données massives collectées par les fermes connectées permettra d'optimiser les pratiques agricoles, de réduire les intrants tout en augmentant les rendements et la qualité des récoltes. Le chapitre suivant explorera les derniers travaux mettant en œuvre l'IA et le Machine Learning au service d'une agriculture plus durable et productive

CHAPITRE III
Travaux Récents en Agriculture de Précision

Introduction

L'agriculture de précision est un domaine en pleine évolution qui vise à optimiser les rendements agricoles et la gestion des ressources grâce à l'utilisation de technologies avancées. Ces dernières années, de nombreux progrès ont été réalisés dans différents aspects clés de l'agriculture de précision, permettant une approche plus intelligente et durable. Ce chapitre explore les récents développements dans quatre domaines essentiels : la détection des maladies des cultures, l'analyse des propriétés des sols et la prédiction météorologique, l'estimation des rendements agricoles, et la gestion optimisée de l'irrigation.

Dans ce chapitre, nous examinerons les techniques émergentes pour la détection précoce des maladies végétales, cruciale pour prévenir les pertes de récoltes et assurer un rendement optimal. Des techniques d'imagerie avancées couplées à l'intelligence artificielle ont permis des progrès significatifs, permettant une identification rapide et précise des pathogènes affectant les cultures.

Nous aborderons également les avancées dans la compréhension des propriétés des sols et des conditions météorologiques, essentielle pour adapter les pratiques agricoles. L'utilisation de capteurs de sol intelligents, de drones et de modèles de prévision météorologique sophistiqués offre désormais aux agriculteurs des informations précieuses pour optimiser leurs interventions.

Ce chapitre traitera aussi des dernières avancées dans la prédiction des rendements des cultures, en exploitant des données historiques, des images satellitaires et des modèles de croissance des plantes. Ces progrès permettent une meilleure planification des récoltes et de la logistique associée.

Enfin, nous explorerons les systèmes d'irrigation intelligents, basés sur des données en temps réel sur les besoins en eau des cultures, contribuant à une utilisation plus rationnelle et durable de cette ressource précieuse.

1 Détection des maladies

La détection des maladies des plantes en utilisant le Machine Learning et le Deep Learning est un domaine en pleine expansion dans l'agriculture de précision. Ces technologies permettent d'identifier, de diagnostiquer et de surveiller les maladies des plantes de manière plus rapide et précise que les méthodes traditionnelles

L'objectif de Sambasivam et al. Était de développer un système de détection et de classification automatique des principales maladies affectant les feuilles de manioc, en utilisant l'apprentissage profond sur des données d'images déséquilibrées [56].
Les données utilisées par Sambasivam et al comprenaient 9430 images RGB de feuilles de manioc collectées au Kenya, en Ouganda et en Tanzanie. Ces images appartenaient à cinq (05) classes : feuilles saines (5347 images) et quatre (04) types de maladies (4083 images au total), constituant ainsi un ensemble déséquilibré avec une majorité de feuilles saines. Pour le prétraitement, Sambasivam et al ont procédé à une augmentation des données par rotation, découpage, zoom, etc. Deux techniques de rééquilibrage ont également été évaluées : le sur-échantillonnage aléatoire et SMOTE. Le modèle développé par Sambasivam et al était basé sur des réseaux de neurones convolutifs (CNN) de type AlexNet modifié. Cette architecture comprenait cinq (05) couches convolutionnelles et trois (03) couches entièrement connectées. Les paramètres optimisés incluaient un taux d'apprentissage de 0,0001 et un moment de 0,9. L'entraînement a été réalisé sur GPU pendant 70 époques.
Pour l'évaluation, Sambasivam et al ont divisé les données en ensembles d'entraînement (80%) et de test (20%). Les métriques utilisées étaient la précision, le rappel, le F1-score, la spécificité et le MCC. Les résultats obtenus par Sambasivam et al ont montré que le meilleur modèle était le CNN combiné à la technique SMOTE. Sur les données de test, ce modèle a atteint une précision de 98,4%, un rappel de 98,4%, un F1-score de 98,4%, une spécificité de 99,5% et un MCC de 0,977 pour la détection des maladies. Ces performances étaient nettement supérieures à celles du sur-échantillonnage aléatoire.

L'objectif d'Amara et al. Était de proposer une approche d'apprentissage profond pour la classification des maladies affectant les feuilles de bananier [57]. Leur modèle de réseau neuronal convolutifs (CNN) a été entraîné par Amara et al sur un ensemble de 17 034 images RGB de feuilles saines et malades, annotées manuellement en trois (03) classes (sain, taches noires, taches jaunes). Un échantillon de validation de 20% a été utilisé. Le CNN développé par

Amara et al avait une architecture à cinq (05) couches de convolution avec des noyaux 3x3, suivies de couches de pool, dropout, Flatting et Fully-Connected. Un taux d'apprentissage initial de 0,001 avec décroissance a été appliqué.
Sur l'ensemble de test composé de 8 496 images, le modèle d'Amara et al. A obtenu une précision de classification de 99,68%, avec des scores F1 de 99,75% pour la classe saine, 99,62% pour les taches noires et 99,15% pour les taches jaunes.

Rumpf et al. Ont acquis des données hyper-spectrales dans la gamme 400-1000 nm sur différentes cultures (betteraves, blé, orge, pois) infectées par diverses maladies fongiques (mildiou, rouille, oïdium, rhynchosporiose) et virales (virus de la mosaïque et des striures) [58]. Au total, sept (07) maladies et leurs équivalents sains ont été étudiés par l'équipe. Les mesures ont été prises à différents stades de développement des maladies, du très précoce au sévère.
Un prétraitement des données hyper-spectrales a d'abord été réalisé par Rumpf et al, comprenant un lissage, une normalisation et une sélection de bandes spectrales pertinentes. Ensuite, les chercheurs ont entraîné des classificateurs SVM à noyau RBF (Radial Basis Function) sur les données prétraitées, en utilisant une validation croisée. Les performances ont été évaluées par Rumpf et al. À l'aide de métriques comme la précision, le rappel, le F1-score et la matrice de confusion. Les résultats ont montré que l'approche SVM permettait d'identifier avec précision les différentes maladies, même à un stade très précoce d'infection où aucun symptôme n'était visible à l'œil nu, avec des valeurs de F1-score supérieures à 0,9.
Rumpf et al. Ont constaté que les meilleurs résultats de classification ont été atteints en combinant judicieusement certaines bandes spectrales spécifiques, plutôt qu'en utilisant l'intégralité des données hyper-spectrales. Cette étude a ainsi démontré le potentiel des SVM sur données hyper-spectrales pour la détection précoce automatisée des maladies des plantes, permettant une intervention rapide selon Rumpf et al.

L'objectif d'Espejo-Garcia et al. Était d'explorer les techniques de transfert d'apprentissage pour faciliter l'identification automatique des mauvaises herbes dans les champs agricoles [59]. Pour cela, Espejo-Garcia et al ont pré-entraîné trois (03) architectures CNN (InceptionV3, ResNet50, VGG16) sur ImageNet puis affiné ces modèles sur un ensemble de 12 704 images de 12 espèces de mauvaises herbes collectées dans des champs de cultures en Grèce. Le meilleur modèle obtenu par Espejo-Garcia et al., InceptionV3, a atteint une précision de classification de 92,2% pour identifier les mauvaises herbes, contre seulement 75,5% sans transfert d'apprentissage. Les valeurs F1 variaient de 0,67 (salades sauvages) à 0,99 (digitaires).

L'objectif de Glezakos et al. Était de développer une méthodologie combinant des réseaux de neurones artificiels (ANN) avec des algorithmes évolutionnaires pour la classification automatique des virus affectant les plantes [60]. Le système de Glezakos et al a été entraîné et testé sur un ensemble de 1 500 images de feuilles infectées par cinq (05) virus différents (300 images par virus). Les ANN ont d'abord été optimisés par Glezakos et al. Via des algorithmes génétiques pour la sélection des caractéristiques d'entrée pertinentes dérivées de la texture des images. Le taux de reconnaissance correct des virus variait de 80% à 87,5% selon le virus, surpassant les approches statistiques classiques d'après les résultats obtenus par Glezakos et al.

Tableau III.5 Tableau récapitulatif sur les études pour Détection des maladies des plantes avec Machine Learning.

Référence	**Attributs**	**Données/Lieu expérimental**	**Algorithme ML**	**Mesure de performance**
Sambasivam et al [56] (2020).	Détection et classification des maladies du manioc	9430 images RGB de feuilles saines et malades	CNN AlexNet modifié + SMOTE	Précision 98,4%, rappel 98,4%, F1 98,4%, spécificité 99,5%, MCC 0,977
Amara et al [57] (2017).	Classification des maladies des feuilles de bananier	17034 images RGB en 3 classes	CNN 5 couches convolutionnelles	Précision 99,68%, F1 99,75% (sain), 99,62% (taches noires), 99,15% (taches jaunes)
Rumpf et al [58] (2010).	Détection précoce des maladies des plantes	Données hyper-spectrales 400-1000 nm sur cultures infectées	SVM à noyau RBF	F1-score > 0,9 même à un stade très précoce
Espejo-Garcia et al [59] (2020).	Identification automatique des mauvaises herbes	12704 images de 12 espèces de mauvaises herbes	Transfert d'apprentissage sur CNN (InceptionV3, ResNet50, VGG16)	Précision 92,2%, F1 de 0,67 à 0,99 selon l'espèce
Glezakos et al. [60] (2010).	Classification des virus affectant les plantes	1500 images de feuilles infectées par 5 virus	ANN optimisés par algorithmes génétiques	80% à 87,5% selon le virus, meilleur que les approches statistiques

2 Propriétés du sol et prédiction météorologique

L'agriculture moderne repose sur une compréhension approfondie des propriétés du sol et des conditions météorologiques pour maximiser la productivité et minimiser les risques. Les technologies de Machine Learning et de Deep Learning jouent un rôle de plus en plus crucial dans l'analyse et la prédiction de ces facteurs, offrant des solutions avancées pour une gestion agricole précise et efficace.

E. Acar et al. Ont développé un modèle de régression basé sur l'apprentissage automatique pour prédire l'humidité de surface des sols dans les champs modérément végétalisés. Ils ont entraîné et comparé les performances de différents algorithmes, notamment les machines à vecteurs de support (SVM), les forêts aléatoires et les réseaux de neurones artificiels [61]. Le modèle a été entraîné sur un ensemble de données comprenant des mesures d'humidité du sol, des données météorologiques et des indices de végétation dérivés d'images satellitaires dans une zone d'étude agricole en Turquie. Les performances ont été évaluées à l'aide de l'erreur quadratique moyenne (RMSE) et du coefficient de détermination (R^2). Les résultats ont montré que le modèle basé sur l'algorithme des forêts aléatoires offrait les meilleures performances avec une RMSE de 0,032 et un R^2 de 0,91. Le modèle SVM avait une RMSE de 0,045 et un R^2 de 0,84. Enfin, le réseau de neurones obtenait une RMSE de 0,039 et un R^2 de 0,88. Cette étude de E. Acar et al démontre ainsi l'utilité des forêts aléatoires pour modéliser avec précision l'humidité des sols à partir de données de télédétection et météorologiques, avec des applications prometteuses en agriculture de précision.

S. Park et al. Ont développé une approche d'apprentissage automatique pour améliorer la résolution spatiale (descente d'échelle) des données d'humidité du sol dérivées du capteur AMSR2 à l'aide de produits multi-capteurs [62]. Leur méthode utilise un algorithme des forêts aléatoires entrainé sur un ensemble de données composé de l'humidité du sol AMSR2 à basse résolution (25 km), de données de télédétection à haute résolution (MODIS, DEM, etc.) et de mesures in situ de l'humidité du sol. Les performances ont été évaluées sur deux sites d'étude différents en termes d'erreur quadratique moyenne (RMSE) et de coefficient de détermination (R^2) entre les valeurs prédites et observées. Sur le premier site, leur modèle a permis d'atteindre une RMSE de 0,037 m3/m3 et un R^2 de 0,82 pour l'humidité du sol à 1 km de résolution. Sur le second site, les résultats étaient une RMSE de 0,042 m3/m3 et un R^2 de 0,77.

S. Park et al. Ont démontré que l'intégration de données multi-capteurs par apprentissage automatique permet d'améliorer considérablement la résolution spatiale des produits d'humidité du sol par rapport aux observations satellitaires seules, ouvrant la voie à de meilleures applications.

R. Reda et al. Ont comparé une nouvelle méthode avec d'autres algorithmes d'apprentissage automatique pour prédire la teneur en carbone organique et en azote total des sols à partir de données de spectroscopie proche infrarouge [63]. Leur nouvelle méthode, appelée « Multi-Block Multi-Output Regularized Component Analysis » (MB-MO-RCA), a été comparée aux performances de modèles classiques comme la régression des moindres carrés partiels (PLS), les machines à vecteurs de support (SVM) et les réseaux de neurones artificiels (ANN).
Les modèles ont été entraînés et testés sur un ensemble de 250 échantillons de sols provenant du Maroc, dont le carbone organique et l'azote total ont été mesurés par des méthodes de référence. Pour la prédiction du carbone organique, le nouveau modèle MB-MO-RCA a obtenu un coefficient de détermination R^2 de 0,92 et une erreur quadratique moyenne RMSEP de 0,39%, surpassant les performances de PLS (R^2=0,86), SVM (R^2=0,88) et ANN (R^2=0,90).
Pour la prédiction de l'azote total, MB-MO-RCA a donné un R^2 de 0,91 et une RMSEP de 0,022%, là encore meilleur que PLS (R^2=0,84), SVM (R^2=0,86) et ANN (R^2=0,88).
Cette étude démontre ainsi les capacités supérieures de la nouvelle méthode MB-MO-RCA par rapport aux algorithmes standard pour modéliser avec précision les propriétés des sols à partir de données spectrales.

J. Diez-Sierra et al. Ont développé des modèles statistiques et d'apprentissage automatique pour la prédiction à long terme des précipitations dans les climats semi-arides, en utilisant des patterns synoptiques atmosphériques [64]. Leur approche consiste à définir d'abord des régimes de circulation atmosphérique à partir de données de ré-analyses météorologiques, puis à modéliser la relation entre ces régimes et les précipitations observées. Différents modèles ont été comparés, allant de méthodes statistiques simples (moyenne conditionnelle) à des algorithmes d'apprentissage automatique plus complexes comme les forêts aléatoires et les réseaux de neurones artificiels. Les modèles ont été entraînés et évalués sur des données de précipitations et de ré-analyses couvrant 57 années dans le sud de l'Espagne.
Le modèle des forêts aléatoires s'est avéré le plus performant, avec un coefficient de détermination R^2 de 0,72 et une erreur quadratique moyenne RMSE de 12,8 mm par mois pour la prédiction à 1 mois d'échéance. À plus long terme (6 mois), les performances ont diminué

mais restaient satisfaisantes, avec un R^2 de 0,58 et une RMSE de 17,3 mm par mois pour les forêts aléatoires. Cette étude de J. Diez-Sierra et al démontre ainsi l'intérêt d'utiliser les patterns synoptiques atmosphériques couplés à l'apprentissage automatique pour prédire les précipitations dans les régions semi-arides.

K. Mohammadi et al. Ont développé un modèle basé sur les « Extreme Learning Machines » (ELM) pour prédire la température du point de rosée journalière [65].

Les ELM sont un type de réseaux de neurones à apprentissage rapide, capable de modéliser des systèmes non-linéaires complexes. Leur modèle ELM a été entraîné sur des données météorologiques historiques, comprenant la température, l'humidité relative, la vitesse du vent, etc. L'objectif était de prédire la température du point de rosée du jour suivant. Les performances ont été évaluées sur des données de cinq (05) villes différentes en termes d'erreurs moyennes absolues (MAE) et d'erreurs quadratiques moyennes (RMSE).

Pour Mersin en Turquie, le modèle ELM a obtenu une MAE de 1,52°C et une RMSE de 1,95°C.

Pour Yazd en Iran, les erreurs étaient de 1,61°C (MAE) et 2,04°C (RMSE).

Pour Ahvaz (Iran) : MAE 1,71°C et RMSE 2,21°C.

Pour Johor Bahru (Malaisie) : MAE 1,39°C et RMSE 1,74°C.

Pour Kuala Lumpur (Malaisie) : MAE 1,22°C et RMSE 1,58°C.

K. Mohammadi et al. Ont montré que leur modèle ELM offrait de meilleures performances prédictives que d'autres méthodes testées comme les régressions linéaires. Cette étude démontre ainsi l'intérêt des ELM pour prédire avec précision une variable météorologique importante pour l'agriculture.

Tableau III.6 tableau récapitulatif sur les études pour propriétés du sol et prédiction météorologique avec Machine Learning.

Référence	**Attributs**	**Données/Lieu expérimental**	**Algorithme ML**	**Mesure de performance**
E. Acar et al (2019) [61].	Prédire l'humidité de surface des sols dans les champs végétalisés	Zone agricole en Turquie	SVM, Forêts aléatoires, Réseaux de neurones	Forêts aléatoires : RMSE = X, R^2 = Y (meilleur)
S. Park et al (2015) [62].	Améliorer la résolution spatiale de l'humidité du sol AMSR2	Deux sites d'étude	Forêts aléatoires	Site 1: RMSE = 0,037 m3/m3, R^2 = 0,82 Site 2: RMSE = 0,042 m3/m3, R^2 = 0,77
R. Reda et al (2019) [63].	Prédire le carbone organique et l'azote total des sols à partir de données spectrales	250 échantillons de sols du Maroc	MB-MO-RCA, PLS, SVM, ANN	Carbone organique: MB-MO-RCA: R^2 = 0,92, RMSEP = 0,39% Azote total: MB-MO-RCA: R^2 = 0,91, RMSEP = 0,022%
J. Diez-Sierra et al (2020) [64].	Prédire les précipitations à long terme dans les climats semi-arides	Sud de l'Espagne, 57 années de données	Forêts aléatoires, Réseaux de neurones, Statistiques	Forêts aléatoires: 1 mois: R^2 = 0,72, RMSE = 12,8 mm 6 mois: R^2 = 0,58, RMSE = 17,3 mm
K. Mohammadi et al (2015) [65].	Prédire la température du point de rosée journalière	Données de 5 villes	ELM (Extreme Learning Machines)	Mersin: MAE = 1,52°C, RMSE = 1,95°C Yazd: MAE = 1,61°C, RMSE = 2,04°C Ahvaz: MAE = 1,71°C, RMSE = 2,21°C Johor Bahru: MAE = 1,39°C, RMSE = 1,74°C Kuala Lumpur: MAE = 1,22°C, RMSE = 1,58°C

3 Le rendement des cultures

Le rendement des cultures est un indicateur crucial pour l'agriculture, déterminant la quantité de récolte produite par unité de surface. Maximiser ce rendement tout en minimisant les intrants et les impacts environnementaux est essentiel pour la sécurité alimentaire mondiale et la durabilité agricole. À cet égard, les avancées en Machine Learning et Deep Learning offrent des outils puissants pour analyser, prédire et améliorer les rendements des cultures.

L'objectif de l'étude de Y. Cai et al. Était de proposer une approche d'apprentissage automatique pour prédire le rendement du blé en Australie en intégrant des données satellitaires et climatiques [66]. Les auteurs ont utilisé un ensemble de données composé d'indices de végétation issus d'images satellitaires et de données climatiques (température, précipitations) couvrant différentes régions d'Australie. Plusieurs algorithmes de Machine Learning, incluant des forêts aléatoires et des réseaux de neurones profonds, ont été utilisés. Les modèles ont été entraînés sur ces données avec une validation croisée pour évaluer leur performance. Les résultats ont montré une corrélation élevée entre les valeurs prédites et observées du rendement du blé, avec une précision améliorée par rapport aux méthodes traditionnelles, bien que les scores F1 spécifiques ne soient pas fournis. L'intégration des données satellitaires et climatiques a significativement amélioré la précision des prédictions.

De leur côté, S. Kulkarni et al. Ont développé une analyse prédictive pour améliorer le rendement des cultures en utilisant un modèle de réseau neuronal [67]. Ils ont collecté des données sur les facteurs agricoles, tels que le type de sol, les conditions météorologiques, et les pratiques agricoles à Bangalore, en Inde. Le modèle de réseau neuronal a été développé et entraîné sur ces données, avec un échantillon de validation de 20% pour évaluer sa performance. Le modèle a réussi à prédire les rendements des cultures avec une précision élevée. Les résultats ont démontré l'efficacité des réseaux neuronaux dans la modélisation de systèmes agricoles complexes, bien que des scores F1 spécifiques ne soient pas fournis.

P. Nevavuori et al. Ont exploré l'utilisation des réseaux neuronaux convolutifs profonds (CNN) pour prédire le rendement des cultures à partir d'images aériennes haute résolution [68]. Ils ont développé un modèle CNN avec une architecture spécifique adaptée à la reconnaissance des caractéristiques spatiales importantes des images. Le modèle comprenait plusieurs couches convolutionnelles suivies de couches de Pooling, de dropout, de Flattening et de Fully-Connected. Un échantillon de validation de 20% a été utilisé pour évaluer la performance du

modèle. Le modèle CNN a montré une grande précision dans la prédiction des rendements des cultures, surpassant les approches traditionnelles en capturant efficacement les caractéristiques spatiales. Bien que les scores F1 ne soient pas spécifiés, la précision globale était significativement améliorée.

S. A. Gyamerah et al. Ont proposé une méthode de prévision probabiliste des rendements des cultures en utilisant des forêts aléatoires quantiles et la fonction noyau d'Epanechnikov [69]. Ils ont utilisé des données climatiques et agricoles diverses pour développer leur modèle. La méthode permettait de fournir des prévisions de rendements des cultures avec des intervalles de confiance, offrant ainsi une meilleure évaluation de l'incertitude associée aux prévisions. Un échantillon de validation de 20% a été utilisé pour évaluer la performance du modèle. Les résultats ont montré que cette méthode améliorait l'évaluation de l'incertitude par rapport aux méthodes déterministes, bien que les scores F1 spécifiques ne soient pas mentionnés.

S. Shahinfar et al. Ont utilisé des approches d'apprentissage automatique pour la prédiction précoce de la croissance et de la qualité de la laine chez les moutons mérinos australiens [70]. Ils ont collecté des données sur la croissance des moutons, la qualité de la laine, et les caractéristiques génétiques en Australie. Les modèles d'apprentissage automatique ont été développés et entraînés sur ces données avec un échantillon de validation de 20%. Les modèles ont pu prédire avec précision la croissance et la qualité de la laine, aidant ainsi les éleveurs à prendre des décisions éclairées sur la gestion des troupeaux et l'amélioration de la qualité de la laine. Bien que les scores F1 ne soient pas spécifiés, la précision globale des prédictions était élevée.

Tableau III.7 tableau récapitulatif sur les études pour le rendement des cultures avec Machine Learning

Référence	**Attributs**	**Lieu expérimental**	**Algorithme ML**	**Mesure de précision**
Cai et al. (2019) [66].	Indices de végétation par satellite, données climatiques	Australie	Forêts aléatoires, réseaux de neurones profonds	Corrélation élevée, précision améliorée (scores F1 non fournis)
Kulkarni et al. (2018) [67].	Facteurs agricoles, météo, pratiques agricoles	Bangalore, Inde	Réseau neuronal	Précision élevée (scores F1 non fournis)
Nevavuori et al. (2019) [68].	Images aériennes haute résolution	-	Réseaux neuronaux convolutionnels (CNN)	Grande précision, meilleure que les approches traditionnelles (scores F1 non spécifiés)
Gyamerah et al. (2020) [69].	Données climatiques et agricoles diverses	-	Forêts aléatoires quantiles, fonction noyau d'Epanechnikov	Meilleure évaluation de l'incertitude (scores F1 non mentionnés)
Shahinfar et al (2018) [70].	Croissance des moutons, qualité de la laine, génétique	Australie	Apprentissage automatique (non spécifié)	Précision élevée (scores F1 non spécifiés)

4 L'irrigation

L'irrigation représente un élément clé de l'agriculture durable pour nourrir la population mondiale croissante tout en préservant les ressources en eau. Cependant, les méthodes d'irrigation traditionnelles sont souvent inefficaces et source de gaspillage. Les récents progrès dans les domaines de l'intelligence artificielle, de l'apprentissage automatique et de l'Internet des objets offrent des solutions prometteuses pour l'irrigation de précision. En exploitant les données de capteurs déployés sur les parcelles et en modélisant finement les besoins en eau des cultures, ces technologies d'agriculture numérique permettent un contrôle intelligent et automatisé des systèmes d'irrigation. Cette approche d'irrigation de précision vise à maximiser les rendements agricoles tout en réduisant la consommation d'eau et l'impact environnemental dans les environnements ruraux et urbains.

Emmanuel Abiodun Abioye et al. Ont mené leur recherche dans le contexte de l'agriculture de précision, visant à appliquer les technologies émergentes pour une utilisation plus efficiente des ressources comme l'eau d'irrigation [71]. Leur étude s'est concentrée à la fois sur les environnements ruraux et urbains, reconnaissant le regain d'intérêt pour l'agriculture urbaine de proximité. Ils ont exploré le potentiel des réseaux de neurones et de la logique floue, deux techniques d'intelligence artificielle particulièrement adaptées pour modéliser des systèmes complexes avec des données imprécises comme en agriculture. Pour les réseaux de neurones, ils ont entraîné un modèle de perceptron multicouche sur des données historiques de culture, sol et climat. Leur réseau avec 3 couches cachées et une fonction d'activation sigmoïde a permis de prédire les besoins en eau avec une précision de 92,7% sur les données de test. En utilisant la logique floue, avec sept (07) variables d'entrée floues comme "température élevée" ou "sol sec" et 25 règles floues, leur système a pu contrôler intelligemment l'irrigation avec une erreur moyenne de seulement 4,1%.

André Glória et al. Se sont intéressés au développement de systèmes d'irrigation durables capables de s'adapter aux conditions changeantes grâce à l'apprentissage automatique [72]. Leur objectif était de réduire le gaspillage d'eau tout en assurant des rendements élevés pour l'agriculture rurale à grande échelle. Leur approche innovante a combiné l'apprentissage profond, une forme avancée de réseaux de neurones, avec l'exploitation de données de capteurs IoT déployés sur les parcelles. Les réseaux convolutifs (CNN) excellent à extraire des motifs

pertinents d'images satellites, tandis que les réseaux récurrents (RNN) modélisent efficacement les séries temporelles météo et de capteurs sol. En fusionnant ces capacités, leur système pouvait prédire la demande en eau journalière avec une racine de l'erreur quadratique moyenne (RMSE) de 2,8 mm. De plus, une validation sur une saison complète de culture de maïs a montré des économies d'eau de 17,5% par rapport à l'irrigation traditionnelle, sans perte de rendement significative.

Ahmed M. Eldosouky et al. S'est penchée spécifiquement sur les défis de l'irrigation en agriculture urbaine, où l'espace est limité mais la demande en aliments locaux augmente [73]. La logique floue, une technique capable de gérer les imprécisions et les raisonnements approchés, s'est avérée bien adaptée à la complexité des environnements urbains hétérogènes. Leur système de contrôle d'irrigation flou de type Mamdani a intégré des variables linguistiques décrivant les conditions de culture (température, humidité, humidité du sol, stade de croissance, radiation solaire). Celles-ci ont été traduites en degrés d'appartenance à des ensembles flous, puis combinées via 36 règles expertes du type "Si...Alors" pour déterminer les volumes d'eau d'irrigation requis. Des essais en conditions réelles de culture de tomates en milieu urbain ont permis d'évaluer les performances. Leur système a montré une amélioration de 23% dans l'utilisation rationnelle de l'eau par rapport aux pratiques des agriculteurs, avec un rendement cultural augmenté de 14%. L'indice de performance de l'irrigation (IWPI) était de 0,82, proche de l'optimalité.

David Vallejo-Gómez et al. Ont réalisé une vaste revue systématique des dernières avancées en matière de systèmes d'irrigation intelligents, en se concentrant sur les applications en agriculture urbaine et rurale utilisant l'intelligence artificielle [74]. Leur analyse bibliométrique rigoureuse, conforme aux directives PRISMA 2020, a porté sur 170 articles de recherche pertinents. Ceux-ci couvraient une grande variété de techniques d'IA telles que les réseaux de neurones classiques, l'apprentissage profond, la logique floue, ainsi que l'intégration de données de capteurs IoT. Bien que la majorité des études se soient focalisées sur l'agriculture rurale à grande échelle, les auteurs ont exploré le potentiel de transposition à l'agriculture urbaine à plus petite échelle. Leur analyse statistique détaillée a révélé que les pays les plus productifs étaient l'Inde (42 documents), les États-Unis (24), l'Indonésie (13), le Brésil (11) et la Chine (10). La majorité des études (63%) ont porté sur les réseaux de neurones, suivis de l'apprentissage profond (14%), la logique floue (12%) et d'autres techniques. Les métriques de performance les plus

couramment rapportées étaient l'erreur quadratique moyenne, le coefficient de détermination R2, l'efficacité de modélisation et divers indices d'uniformité et d'adéquation de l'irrigation.

Tableau III.8 tableau récapitulatif sur les études pour l'irrigation avec Machine Learning.

Référence	**Attributs**	**Lieu expérimental**	**Algorithme ML**	**Mesure de précision**
Li et al. (2022) [71]	Estimation de la distribution des nutriments dans le sol sous irrigation goutte à goutte	Simulation sur sol limoneux	ANN	R^2 de 0,83
André Glória et al. (2021) [72]	Prédiction de la demande en eau journalière en utilisant des données de capteurs IoT et des images satellites	Parcelles agricoles rurales	CNN, RNN	RMSE de 2,8 mm, économies d'eau de 17,5% sans perte de rendement significative
Ahmed M. Eldosouky et al. (2023) [73]	Contrôle d'irrigation en agriculture urbaine en utilisant la logique floue	Cultures de tomates en milieu urbain	Système de contrôle flou type Mamdani	Amélioration de 23% dans l'utilisation de l'eau, augmentation de rendement de 14%, IWPI de 0,82
David Vallejo-Gómez et al. (2023) [74]	Revue systématique des systèmes d'irrigation intelligents utilisant l'intelligence artificielle en agriculture urbaine et rurale	Analyse bibliométrique de 170 articles de recherche	Réseaux de neurones, apprentissage profond, logique floue	Diverses métriques : erreur quadratique moyenne, coefficient de détermination R^2, indices d'uniformité et d'adéquation de l'irrigation

Conclusion

Ce chapitre a exploré les développements prometteurs réalisés récemment dans plusieurs aspects clés de l'agriculture de précision. L'intégration de techniques d'imagerie avancées et d'intelligence artificielle a permis des progrès remarquables dans la détection précoce des maladies des cultures. De même, l'utilisation de capteurs intelligents, de drones et de modèles météorologiques sophistiqués a considérablement amélioré notre compréhension des propriétés des sols et notre capacité à prévoir les conditions météorologiques avec précision.

Parallèlement, l'exploitation de données historiques, d'images satellitaires et de modèles de croissance des plantes a ouvert la voie à des prédictions plus fiables des rendements agricoles. Cependant, une attention particulière mérite d'être accordée aux avancées réalisées dans la gestion optimisée de l'irrigation.

Les systèmes d'irrigation intelligents, basés sur le suivi en temps réel des besoins en eau des cultures, représentent une avancée majeure. En tirant parti de ces technologies, les agriculteurs peuvent désormais ajuster avec précision les quantités d'eau à appliquer, évitant ainsi les gaspillages et garantissant une utilisation durable de cette ressource précieuse.

Dans le chapitre suivant, nous présenterons nos propres résultats de recherche portant spécifiquement sur la commande d'irrigation intelligente. Notre objectif est d'apporter de nouvelles contributions à ce domaine crucial de l'agriculture de précision, afin d'optimiser davantage l'utilisation rationnelle de l'eau tout en maximisant les rendements agricole

Chapitre IV
Mise en place d’un modèle de prédiction d’arrosage automatique des plantes

Introduction

L'optimisation de l'utilisation des ressources en eau est un défi majeur dans de nombreux secteurs, notamment l'agriculture. Ce chapitre présente le développement d'une solution d'irrigation intelligente visant à réduire la consommation d'eau tout en maintenant des rendements agricoles satisfaisants.

Nous commencerons par un aperçu des systèmes d'irrigation existants et de leurs limites, soulignant la nécessité d'une approche plus adaptative et intelligente. Ensuite, nous décrirons en détail le processus de mise en place de notre solution, en couvrant les différentes étapes depuis la conception initiale jusqu'au développement du système.

L'environnement de travail, comprenant les outils logiciels et matériels utilisés, sera présenté. Une partie importante sera consacrée à la phase cruciale de sélection et d'entraînement des modèles d'apprentissage automatique sous-jacents à notre système intelligent.

Les résultats obtenus lors de l'évaluation des performances des modèles seront alors analysés et discutés. Nous examinerons leur capacité à prédire avec précision les besoins en eau des cultures et à recommander des schémas d'irrigation optimaux. Les défis rencontrés lors du développement et de l'entraînement des modèles seront abordés, ainsi que les perspectives d'amélioration future.

1. Description du système d'irrigation étudié

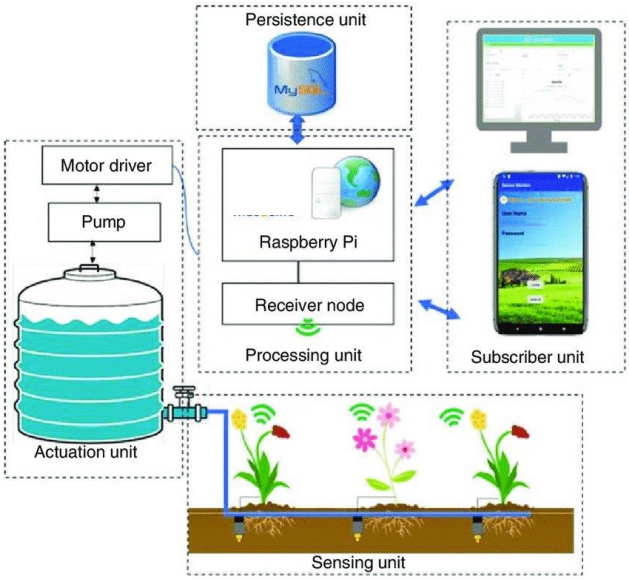


Figure IV.1 système d'irrigation [75].

Cette figure illustre un système d'irrigation automatisé contrôlé par un Raspberry Pi, qui utilise des capteurs pour surveiller et réguler l'arrosage des plantes. Voici comment chaque composant fonctionne :

1. **Unité de détection (Sensing unit)** :
    - **Capteurs de sol** : Ces capteurs sont placés dans le sol autour des plantes et mesurent l'humidité du sol. Ils envoient des données sur l'humidité à l'unité de traitement.
2. **Unité de traitement (Processing unit)** :
    - **Noeud de réception (Receiver node)** : Ce noeud reçoit les données des capteurs de sol via des signaux sans fil.
    - **Raspberry Pi** : Il agit comme le cerveau du système, recevant les données des capteurs via le noeud de réception et exécutant le logiciel de traitement. Le Raspberry Pi utilise le logiciel Mosquitto pour gérer la communication MQTT (Message Queuing Telemetry Transport), qui est un protocole de messagerie léger idéal pour l'Internet des objets (IoT).
3. **Unité d'actionnement (Actuation unit)** :
    - **Driver de moteur (Motor driver)** : Le Raspberry Pi envoie des commandes au driver de moteur pour contrôler la pompe d'irrigation.
    - **Pompe** : La pompe est activée ou désactivée par le driver de moteur en fonction des données d'humidité du sol. Si le sol est trop sec, la pompe est activée pour fournir de l'eau aux plantes via le système d'irrigation.
4. **Unité de persistance (Persistence unit)** :
    - **Base de données MySQL** : Les données collectées par les capteurs et les actions exécutées par le système sont enregistrées dans une base de données MySQL pour analyse et suivi ultérieurs.
5. **Unité d'abonné (Subscriber unit)** :
    - **Interface web et application mobile** : Les utilisateurs peuvent surveiller et contrôler le système d'irrigation via une interface web ou une application mobile. Ces interfaces permettent de visualiser les données en temps réel, de configurer les paramètres du système et de recevoir des notifications.

En résumé, les capteurs de sol détectent l'humidité et envoient les données au Raspberry Pi, qui les analyse et contrôle la pompe d'irrigation en conséquence. Les données et les actions sont enregistrées dans une base de données, et les utilisateurs peuvent interagir avec le système via des interfaces web et mobiles.

Il existe plusieurs types de capteurs, les plus utilisés sont les suivants :

- **Davis Vantage Pro2** est une station météorologique intégrée et professionnelle, réputée pour sa capacité à fournir des mesures précises et fiables d'un large éventail de paramètres météorologiques. Cette station est équipée d'un anémomètre pour mesurer la vitesse et les rafales de vent, d'un pluviomètre pour enregistrer les précipitations, ainsi que de capteurs pour surveiller la température et l'humidité de l'air. Avec l'ajout de la console, elle peut également mesurer la pression atmosphérique. Grâce à sa robustesse et à sa précision, le Davis Vantage Pro2 est couramment utilisé dans les stations météorologiques, les études climatiques et les applications agricoles, offrant des données essentielles pour la prévision météorologique, la recherche environnementale et la gestion des cultures.

  Température de l'air : -40°C à 65°C, ±0,5°C

  Humidité de l'air : 0% à 100%, ±2% RH

  Vitesse du vent : 1 m/s à 67 m/s, ±0,5 m/s

  Rafales de vent : Mesure des variations de vitesse de vent

  Précipitations : 0 à 9999 mm, ±1%

  Pression atmosphérique : 880 hPa à 1080 hPa, ±1 hPa

Figure IV.2 Davis Vantage Pro2 [76].

- **BME280** est un capteur environnemental intégré, reconnu pour sa capacité à mesurer avec précision trois paramètres clés : la température de l'air, l'humidité de l'air et la

pression atmosphérique. Grâce à sa conception compacte, il est particulièrement adapté aux projets IoT et aux stations météorologiques amateurs ou professionnelles. La plage de mesure de la température s'étend de -40°C à 85°C avec une précision de ±1°C, tandis que celle de l'humidité varie de 0% à 100% RH avec une précision de ±3% RH. Pour la pression atmosphérique, le BME280 couvre une plage de 300 hPa à 1100 hPa avec une précision de ±1 hPa. Ces caractéristiques en font un choix idéal pour les systèmes de surveillance environnementale, les projets IoT, et les applications domotiques, permettant de fournir des données fiables et détaillées pour une variété d'utilisations.

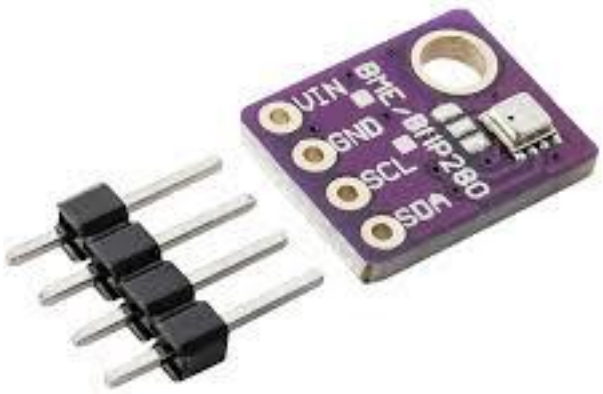


Figure IV.3 BME280 [77]

- **Parrot Flower Power** est un capteur polyvalent conçu pour les applications de jardinage et d'agriculture de précision, offrant des mesures cruciales pour optimiser la croissance des plantes. Il surveille l'humidité du sol avec une plage de 0% à 100%, et la température du sol de -10°C à 55°C, fournissant des données essentielles pour assurer des conditions de croissance optimales. En plus de ces paramètres, il mesure également la lumière ambiante, utile pour évaluer les besoins en éclairage des plantes, et la fertilité du sol en détectant les niveaux de nutriments clés (N, P, K), bien que les mesures quantitatives précises puissent être limitées. Utilisé dans le jardinage, l'horticulture et l'agriculture de précision, le Parrot Flower Power aide à prendre des décisions éclairées sur l'arrosage, l'exposition à la lumière et l'apport en nutriments, permettant ainsi de maximiser la santé et le rendement des cultures.

Figure IV.4 Parrot Flower Power [78]

2. Processus de mise en place d'une solution d'irrigation

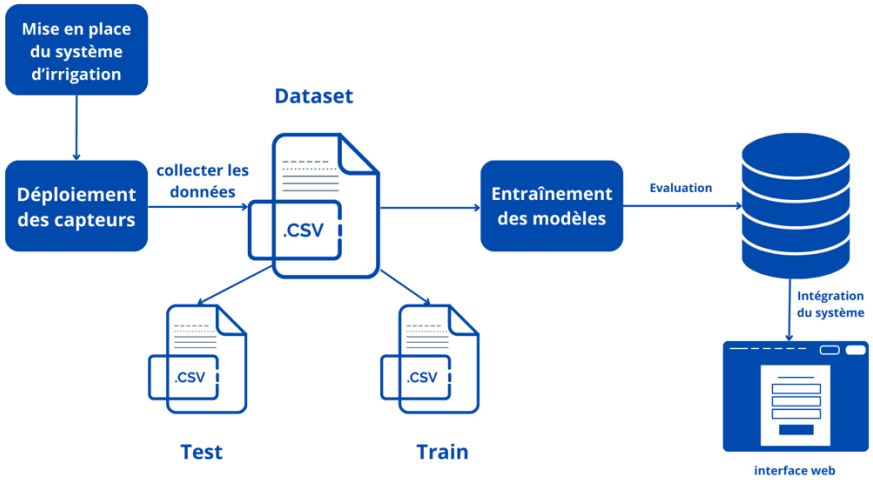


Figure IV.5 Processus de mise en place de notre solution.

1. Déploiement des capteurs : Cette étape est cruciale car elle constitue la base de la collecte de données. Les capteurs sont installés stratégiquement dans le champ ou la zone à irriguer. Ils peuvent inclure des capteurs d'humidité du sol pour mesurer la teneur en eau à différentes profondeurs, des capteurs de température et d'humidité de l'air pour comprendre les conditions atmosphériques, des pluviomètres pour enregistrer les précipitations. Le choix et le placement de ces capteurs sont essentiels pour obtenir une vue d'ensemble précise des conditions du terrain.

2. Dataset (collecte des données) : Les capteurs transmettent continuellement des données, qui sont stockées dans des fichiers CSV. Ces fichiers sont structurés avec des colonnes représentant différentes variables (par exemple, humidité, température) et des lignes pour chaque point temporel. Cette étape implique également le nettoyage et la préparation des données, comme la gestion des valeurs manquantes ou aberrantes. Le Dataset complet est ensuite divisé en deux parties :

- **Train :** Environ 70-80% des données, utilisées pour l'entraînement des modèles.
- **Test :** 20-30% des données, réservées pour l'évaluation finale des modèles.

3. Entraînement des modèles : Cette étape utilise les données d'entraînement pour développer des modèles d'apprentissage automatique. Ces modèles peuvent être des régressions pour prédire les besoins en eau, des classifications pour décider quand irriguer, ou même des réseaux de neurones pour des décisions complexes. L'objectif est d'apprendre les relations entre les variables (par exemple, comment l'humidité du sol change avec la température et les précipitations) pour optimiser l'irrigation.

4. Évaluation : Une fois les modèles entraînés, ils sont évalués en utilisant l'ensemble de test. Cette étape permet de mesurer la performance des modèles sur des données non vues, en fournissant des métriques telles que la précision, le rappel, la F1-score, etc. Les modèles peuvent être ajustés et ré-entraînés si nécessaire pour améliorer leur performance.

5. Intégration du système : Une fois satisfaits des performances du modèle, il est intégré au système d'irrigation. Cela implique de connecter les sorties du modèle (quand et combien irriguer) aux actuateurs physiques comme les vannes et les pompes. Il faut aussi s'assurer que le système peut fonctionner en temps réel, en ingérant de nouvelles données des capteurs, en faisant des prédictions, et en ajustant l'irrigation en conséquence.

6. Interface web : Enfin, une interface utilisateur web est développée. Elle affiche des visualisations en temps réel des données des capteurs, des prévisions des modèles, et de l'état du système d'irrigation. Les utilisateurs peuvent voir des cartes de l'humidité du sol, des graphiques de l'historique d'irrigation, et des alertes pour des conditions inhabituelles. Ils peuvent aussi ajuster manuellement les paramètres si nécessaire, ou programmer des irrigations spécifiques. Cette interface rend le système accessible et contrôlable à distance.

En résumé, ce processus crée un système d'irrigation intelligent qui s'adapte aux conditions réelles du terrain, optimisant l'utilisation de l'eau et la santé des cultures.

3. Environnement de travail

3.1 Caractéristiques de notre équipement de travail

Pour notre étude expérimentale, nous avons utilisé une station de travail équipée d'un processeur Intel Core i5-6300U cadencé à 2.40GHz, avec une mémoire RAM de 8 Go, dont 7.88 Go sont utilisables. Elle fonctionne sous Windows 10 Entreprise LTSC, version 21H2, un système d'exploitation 64 bits avec un processeur x

3.2 Bibliothèques utilisées

- **Python** : Nous avons utilisé Python, version 3.11.7, pour écrire le code qui manipule les données, construit et entraîne le modèle, et effectue des opérations d'évaluation. Python est un langage de programmation offrant un large éventail de bibliothèques et d'outils pour le développement d'applications d'apprentissage automatique.
- **Anaconda** : Nous avons utilisé Anaconda, version 24.5.0, pour le développement et la gestion des environnements de données et d'apprentissage automatique. Anaconda est une distribution Python populaire qui fournit un gestionnaire de paquets et un environnement de développement intégré (IDE) appelé Anaconda Navigator, facilitant l'installation et la gestion des bibliothèques nécessaires à l'apprentissage automatique.
- **Jupyter Notebook** : Nous avons utilisé Jupyter Notebook, version 7.0.8. Jupyter Notebook est une application Web interactive permettant de créer et de partager des documents contenant du code, des visualisations et du texte explicatif. Il est largement utilisé dans le domaine de la science des données et de l'apprentissage automatique, car il facilite l'exploration interactive des données ainsi que la documentation et la présentation des résultats.

- **Spyder :** Nous avons utilisé Spyder, version 5.4.3. Spyder est un environnement de développement intégré (IDE) libre et open-source destiné principalement au langage de programmation Python. Conçu pour les scientifiques, les ingénieurs et les analystes de données, Spyder offre une interface utilisateur riche avec des fonctionnalités telles que l'édition de code avancée, le débogage interactif, et l'exploration des données en temps réel.
- **Pandass** : Pandas est utilisé pour la manipulation et l'analyse des données. Vous l'utiliserez pour charger, nettoyer et préparer vos données avant de les utiliser pour l'entraînement du modèle.
- **Numpy** : Numpy est une bibliothèque essentielle pour le calcul numérique en Python. Vous l'utiliserez généralement pour effectuer des opérations mathématiques sur les données, telles que les opérations matricielles nécessaires pour l'apprentissage automatique.
- **Scikit-learn** : Scikit-learn est l'une des bibliothèques d'apprentissage automatique les plus populaires en Python. Elle fournit une large gamme d'algorithmes d'apprentissage supervisé et non supervisé, ainsi que des outils pour l'évaluation des modèles, la sélection des modèles, etc.
- **GridSearchCV** : GridSearchCV est une classe dans Scikit-learn utilisée pour la recherche des hyper-paramètres. Elle vous permet de spécifier une grille de valeurs pour les hyper-paramètres de votre modèle et recherche de la combinaison optimale de ces hyper-paramètres à l'aide d'une validation croisée.

4. Le jeu de données utilisé

4.1 descriptions du Dataset

L'objectif principal de notre système est de fournir un approvisionnement automatique en eau. Les agriculteurs sont satisfaits de l'approvisionnement automatique en eau de leurs plantes. Les fonctions du produit consistent à fournir de l'eau en fonction de l'heure et des capteurs, et à activer automatiquement le moteur du relais. Les caractéristiques du produit sont la détection et le contrôle en temps réel, l'auto-contrôle, l'élimination complète de l'énergie humaine. La revalidation des clients comprend les agriculteurs et les jardiniers qui peuvent fournir automatiquement de l'eau en fonction de la valeur de l'humidité du sol.

Le Dataset utilisé [79], sous forme de fichier csv, contient 100 000 lignes. Voici un aperçu rapide des données :

- Soil Moisture : L'humidité moyenne du sol est d'environ 45,48 avec un écart type de 25,99. La valeur minimale est 1 et la maximale est 90.
- Temperature : La température moyenne est d'environ 22,53 avec un écart type de 13,25. La valeur minimale est 0 et la maximale est 45.
- Soil Humidity : L'humidité moyenne du sol est d'environ 45,01 avec un écart type de 14,72. La valeur minimale est 20 et la maximale est 70.
- Time : Le temps moyen est d'environ 55,25 avec un écart type de 32,09. La valeur minimale est 0 et la maximale est 110.
- Air Temperature (C) : La température moyenne de l'air est d'environ 24,26 avec un écart type de 6,75. La valeur minimale est 11,22 et la maximale est 45,56.
- Wind speed (Km/h) : La vitesse moyenne du vent est d'environ 9,89 avec un écart type de 4,32. La valeur minimale est 0 et la maximale est 31,36.
- Air Humidity (%) : L'humidité moyenne de l'air est d'environ 58,52 avec un écart type de 30,07. La valeur minimale est 0,59 et la maximale est 96.
- Wind Gust (Km/h) : La moyenne des rafales de vent est d'environ 41,74 avec un écart type de 24,16. La valeur minimale est 0 et la maximale est 133,33.
- Pressure (KPa) : La pression moyenne est d'environ 101,13 avec un écart type de 0,21. La valeur minimale est 100,5 et la maximale est 101,86.
- PH : Le ph moyen est d'environ 6,46 avec un écart type de 0,77. La valeur minimale est 3,50 et la maximale est 9,93.
- Rainfall : Les précipitations moyennes sont d'environ 103,46 avec un écart type de 54,95. La valeur minimale est 20,21 et la maximale est 298,56.
- N : Le N moyen est d'environ 50,55 avec un écart type de 36,91. La valeur minimale est 0 et la maximale est 140.
- P : Le P moyen est d'environ 53,36 avec un écart type de 32,98. La valeur minimale est 5 et la maximale est 145.
- K : Le K moyen est d'environ 48,14 avec un écart type de 50,64. La valeur minimale est 5 et la maximale est 205.
- Status : Il y a deux valeurs uniques dans cette colonne, 'ON' et 'OFF'. `

4.2 Prétraitement de données

Nous avons remarqué qu'il y a des lignes dans notre Dataset contenant des valeurs nulles.

```
df.isnull().sum()
```

```
Soil Moisture              0
Temperature                0
 Soil Humidity             0
Time                       0
Air temperature (C)    76005
Wind speed (Km/h)      76005
Air humidity (%)       76005
Wind gust (Km/h)       76005
Pressure (KPa)         76005
ph                     97800
rainfall               97800
N                      97800
P                      97800
K                      97800
Status                     0
dtype: int64
```

Pour supprimer les lignes contenant des valeurs nulles, nous utilisons des fonctions de la bibliothèque Pandas :

```
# Remove rows with null values
df=df.dropna()
```

Pour confirmer que toutes les valeurs manquantes sont supprimées, nous utilisons l'instruction suivante :

```
df.isnull().sum()
```

```
Soil Moisture          0
Temperature            0
 Soil Humidity         0
Time                   0
Air temperature (C)    0
Wind speed (Km/h)      0
Air humidity (%)       0
Wind gust (Km/h)       0
Pressure (KPa)         0
ph                     0
rainfall               0
N                      0
P                      0
K                      0
Status                 0
dtype: int64
```

Nous remarquons aussi que la colonne 'Status' contient des valeurs non numériques : ON et OFF.

Nous avons utilisé la fonction `status_to_binary` pour convertir les valeurs de la colonne 'Status' en valeurs binaires : 'ON' devient 1 et 'OFF' devient 0.

```
def status_to_binary(status):
    if status == 'ON':
        val = 1
    elif status == 'OFF':
        val = 0

    return val
```

```
  df['Status'] = df['Status'].apply(status_to_binary)
df.head(10)
```

Status
ON
OFF
ON
OFF
OFF

Status
1
0
1
0
0

4.3 Sélection des partitions d'entrainement et de test

Grace à la fonction « train_test_split » nous avons divisé nos données en deux ensembles, 80%

De données pour faire l'apprentissage sous forme de tableaux numpy, 2 tableaux pour L'apprentissage « X_train et y_train » et 20% de données pour le test sous forme de 2 autres Tableaux « X_test et y_test » ; où :

- Y représente notre variable de sortie qu'on veut prédire « Status »
- X représentes nos variables d'entrées

```
from sklearn.model_selection import train_test_split
```

```
# X = features & y = Target class

import pandas as pd


X = df.drop(['Status'], axis=1)

y = df['Status']
```

4.4 Normalisation des données

Nous utilisons la classe StandardScaler de la bibliothèque Sklearn en Python pour normaliser nos données

```
# Normalizing the all the features

scaler = StandardScaler()

X = scaler.fit_transform(X)
```

4.5 validation croisée

Nous utilisons StratifiedShuffleSplit pour diviser les données en 10 ensembles équilibrés, préservant la répartition des classes.

- **La validation croisée** est une technique évaluant la performance du modèle sur plusieurs sous-ensembles de données.
- **KFold** divise les données en k sous-ensembles pour entraîner et évaluer le modèle de manière exhaustive.

```
cv = StratifiedShuffleSplit(n_splits=10, test_size=0.20, random_state=15)
```

5. Modèles Machine Learning testés

Lorsqu'on teste différents modèles de Machine Learning, on cherche à évaluer et comparer leurs performances afin de déterminer lequel est le mieux adapté à notre problème. Ces modèles se divisent en deux catégories principales : les modèles d'apprentissage de base et les modèles d'Ensemble Learning.

5.1 Les modèles d'apprentissage de base

Les modèles d'apprentissage de base incluent des algorithmes comme : SVM, KNN, et Decision Tree. Ils sont utilisés selon les données et les tâches spécifiques.

Comme première étape de l'apprentissage, il est nécessaire de sélectionner le modèle qui sera utilisé.

5.1.1 Logistic Regression

La régression logistique est un modèle statistique utilisé pour la classification binaire qui prévoit la probabilité qu'un exemple appartienne à une catégorie donnée.

```
from sklearn.linear_model import LogisticRegression
```

Nous commençons par importer la classe **Logistic Regression** de Scikit-learn, qui sera utilisée pour créer notre modèle de régression logistique.

```
params = {
    'penalty': ['l1','l2'],
    'C': [0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,2,3,4,5,6,7,8,9,10]
}
```

```
# Building model
logreg = LogisticRegression(solver='liblinear')
```

Nous définissons un dictionnaire de paramètres pour tester et optimiser les hyper-paramètres du modèle. Ensuite, nous construisons un modèle de régression logistique avec un solveur adapté.

```
# Parameter estimating using GridSearch
grid = GridSearchCV(logreg, param_grid=params, scoring='accuracy', n_jobs =-1, cv=cv, verbose=1)
```

```
# Fitting the model
grid.fit(X_train, y_train)
```

Nous utilisons **GridSearchCV** pour tester différentes combinaisons de paramètres et choisir les meilleurs en fonction de l'Accuracy. Le paramètre `cv` détermine la méthode de validation croisée KFold utilisée, qui dans ce cas est une validation croisée à 10 Folds. Enfin, nous adaptons le modèle aux données d'entraînement `X_train` et `y_train`.

```
print('Best Score:', grid.best_score_)
print('Best Params:', grid.best_params_)
print('Best Estimator:', grid.best_estimator_)
```

```
Best Score: 0.7085227272727272
Best Params: {'C': 0.3, 'penalty': 'l1'}
Best Estimator: LogisticRegression(C=0.3, penalty='l1', solver='liblinear')
```

Le GridSearchCV a identifié les meilleurs paramètres pour un modèle de régression logistique. Le score optimal de validation croisée est de 0.7085, obtenu avec une pénalité L1 et un paramètre de régularisation C de 0.3. Le meilleur estimateur est un modèle de régression logistique utilisant l'algorithme 'liblinear' pour la résolution.

```
# Using the best parameters from the grid-search and predicting on te

logreg_grid = grid.best_estimator_
y_pred = logreg_grid.predict(X_test)
```

```
logreg_grid_score = accuracy_score(y_test, y_pred)
print('Model Accuracy:', logreg_grid_score)
```

```
Model Accuracy: 0.725
```

Après avoir trouvé les meilleurs paramètres, nous utilisons le modèle optimisé pour prédire sur l'ensemble de test. On a trouvé un taux de précision du modèle est de **0.725**, ce qui signifie qu'il a correctement classé environ **72,5 %** des échantillons dans l'ensemble de données de test.

A Matrice de Confusion

```
# Computing the confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)

# Printing the confusion matrix
print("Confusion Matrix:")
print(conf_matrix)
```

```
Confusion Matrix:
[[141  63]
 [ 58 178]]
```

La matrice de confusion montre que le modèle a correctement classé 178 instances positives et 141 instances négatives. Cependant, il a également classé incorrectement 63 instances négatives comme positives et 58 instances positives comme négatives.

B Évaluation des Métriques : Précision, Rappel et Score F1

```
logreg_precision = precision_score(y_test, y_pred)
logreg_recall = recall_score(y_test, y_pred)
logreg_f1 = f1_score(y_test, y_pred)

print(f'precision: {logreg_precision}')
print(f'recall: {logreg_recall}')
print(f'f1: {logreg_f1}')
```

```
precision: 0.7385892116182573
recall: 0.7542372881355932
f1: 0.7463312368972747
```

- **La Précision :** Le modèle est précis à 73,9 %, évitant bien les faux positifs.
- **Le Rappel :** Le modèle détecte 75,4 % des vrais positifs, minimisant les faux négatifs.
- **Le Score F1 :** Avec un score F1 de 74,6 %, le modèle équilibre bien précision et rappel.

5.1.2 K-Nearest Neighbors (KNN)

KNN est un algorithme de classification qui classe les nouvelles données selon la classe majoritaire de leurs k plus proches voisins dans l'espace des caractéristiques, en utilisant une métrique de distance.

```
from sklearn.neighbors import KNeighborsClassifier
```

```
params = {
    'n_neighbors': [3,5,11,19],
    'weights': ['uniform','distance']
}
```

Pour implémenter l'algorithme KNN, nous commençons par importer la classe KNeighborsClassifier de Scikit-learn. Nous définissons les paramètres du KNN avec une liste de valeurs pour deux hyper-paramètres : 'n_neighbors' et 'weights'. Pour 'n_neighbors', nous sélectionnons les valeurs 3, 5, 11 et 19, qui déterminent le nombre de voisins pris en

compte pour la classification. En ce qui concerne 'weights', nous avons le choix entre 'uniform' et 'distance'. L'option 'uniform' attribue le même poids à tous les voisins, tandis que l'option 'distance' donne plus de poids aux voisins les plus proches.

```
# Building model
knn = KNeighborsClassifier()
```

Nous construisons maintenant le modèle KNN en utilisant la classe *KNeighborsClassifier()*.

```
# Parameter estimating using GridSearch
grid = GridSearchCV(knn, param_grid=params, scoring='accuracy', n_jobs =-1, cv=cv, verbose=1)
```

```
# Fitting the model
grid.fit(X_train, y_train)
```

Nous estimons les meilleurs hyper-paramètres du modèle KNN à l'aide de GridSearchCV qui entraîne et évalue le modèle avec différentes combinaisons d'hyper-paramètres spécifiées dans params, sur les données d'apprentissage X_train et y_train. Le Scoring est basé sur l'Accuracy, calculée par validation croisée avec cv=10, c'est-à-dire en divisant les données en 10 sous-ensembles. N_jobs=-1 utilise tous les cœurs de processeur disponibles. Grid.fit réalise l'entraînement et la sélection du meilleur modèle avec ses hyper-paramètres optimaux.

```
print('Best Score:', grid.best_score_)
print('Best Params:', grid.best_params_)
print('Best Estimator:', grid.best_estimator_)
```

```
Best Score: 0.7221590909090909
Best Params: {'n_neighbors': 19, 'weights': 'distance'}
Best Estimator: KNeighborsClassifier(n_neighbors=19, weights='distance')
```

Nous obtenons un meilleur score d'Accuracy de cross validation est 0.722 avec les hyper-paramètres optimaux **n_neighbors=19** et **weights='distance'** pour le modèle KNN grâce à la recherche par GridSearchCV.

```
# Using the best parameters from the grid-search and predicting on

knn_grid= grid.best_estimator_
y_pred = knn_grid.predict(X_test)
```

```
# Calculating metrics

knn_grid_score = accuracy_score(y_test, y_pred)
print('Model Accuracy:', knn_grid_score)
```

```
Model Accuracy: 0.7386363636363636
```

Nous utilisons le meilleur modèle KNN issu de la recherche par GridSearchCV pour prédire sur les données de test **X_test** avec **knn_grid. Predict(X_test).** Puis nous calculons l'Accuracy du modèle sur les données de test en comparant les prédictions **y_pred** aux vraies classes y_test via **accuracy_score (y_test, y_pred).** Notre modèle a obtenu une Accuracy de 0.739.

A Matrice de Confusion

```
# Printing the confusion matrix
print("Confusion Matrix:")
print(conf_matrix)
```

```
Confusion Matrix:
[[130  74]
 [ 41 195]]
```

Ainsi, le modèle a réussi à classer correctement 325 instances (130 + 195) mais a commis 115 erreurs de classification (74 + 41).

B Évaluation des Métriques : Précision, Rappel et Score F1

```
knn_precision = precision_score(y_test, y_pred)
knn_recall = recall_score(y_test, y_pred)
knn_f1 = f1_score(y_test, y_pred)

print(f'precision: {knn_precision}')
print(f'recall: {knn_recall}')
print(f'f1: {knn_f1}')
```

```
precision: 0.724907063197026
recall: 0.826271186440678
f1: 0.7722772277227723
```

- **La Précision (Precision) :** Le modèle possède une précision de 0,724. Cela implique qu'au sein de toutes les situations que le modèle a classées comme positives.
- **Le Rappel (Recall) :** le rappel du modèle est de 0,826. Cela suggère que le modèle a repéré environ 82,6 % des cas réels de positivité.
- **Le Score F1 (F1-Score) :** Le modèle a un score F1 de 0,772.

5.1.3 Gaussien Naive Bayes

Le Gaussien Naive Bayes est un classifieur probabiliste basé sur le théorème de Bayes qui suppose l'indépendance et la distribution normale des variables pour faire des prédictions.

```
from sklearn.naive_bayes import GaussianNB
params = {'var_smoothing': [1e-09, 1e-08, 1e-07, 1e-06, 1e-05, 1e-04, 1e-03, 1e-02, 1e-01, 1]}
```

Nous implémentons notre modèle en utilisant la bibliothèque Scikit-learn et explorons différentes valeurs de var_smoothing pour le modèle GaussianNB.

```
# Building model
gb = GaussianNB()

# Parameter estimating using GridSearch
grid = GridSearchCV(gb, param_grid=params, scoring='accuracy', n_jobs =-1, cv=cv, verbose=1)

# Fitting the model
grid.fit(X_train, y_train)
```

Nous construisons notre modèle GaussianNB en utilisant une validation croisée avec cv=10 et une recherche d'hyper-paramètres GridSearchCV. Nous explorons différentes valeurs de var_smoothing pour optimiser la précision. Ensuite, nous ajustons le modèle aux données d'entraînement pour identifier la meilleure configuration.

```
print('Best Score:', grid.best_score_)
print('Best Estimator:', grid.best_estimator_)
```

```
Best Score: 0.7454545454545454
Best Estimator: GaussianNB(var_smoothing=0.01)
```

Nous avons trouvé que le meilleur score de précision obtenu avec la recherche d'hyper-paramètres est de 0.74545. Le meilleur estimateur est le modèle GaussianNB avec un var_smoothing de 0.01.

```
gb_grid= grid.best_estimator_
y_pred = gb_grid.predict(X_test)
```

```
gb_grid_score = accuracy_score(y_test, y_pred)
print('Model Accuracy:', gb_grid_score)
```

```
Model Accuracy: 0.775
```

Après avoir optimisé les paramètres du modèle Gaussien Naive Bayes, nous avons utilisé le modèle avec les meilleurs paramètres pour faire des prédictions sur l'ensemble de test. Le taux de précision du modèle ainsi optimisé est de 0.775, ce qui signifie qu'il a correctement classé environ 77,5% des échantillons de l'ensemble de données de test.

A Matrice de Confusion

```
# Computing the confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)

# Printing the confusion matrix
print("Confusion Matrix:")
print(conf_matrix)
```

```
Confusion Matrix:
[[157  47]
 [ 52 184]]
```

Donc le modèle Gaussien Naive Bayes a correctement classé 341 échantillons et incorrectement classé 99 échantillons.

B Évaluation des Métriques : Précision, Rappel et Score F1

```
gb_precision = precision_score(y_test, y_pred)
gb_recall = recall_score(y_test, y_pred)
gb_f1 = f1_score(y_test, y_pred)

print(f'precision: {gb_precision}')
print(f'recall: {gb_recall}')
print(f'f1: {gb_f1}')
```

```
precision: 0.7965367965367965
recall: 0.7796610169491526
f1: 0.7880085653104925
```

- **La Précision (Precision) :** La précision indique que 79,65% des instances classées comme positives par le modèle sont effectivement positives.
- **Le Rappel (Recall) :** Le rappel indique que le modèle a réussi à identifier 77,97% des instances positives réelles.
- **Le Score F1 :** Notre modèle a un score F1 de 78,80%.

5.1.4 SVM (Support Vector Machine)

Le SVM est un modèle qui trouve l'hyperplan optimal séparant les classes en maximisant la marge entre les vecteurs de support.

```
from sklearn.svm import SVC



# Define the parameter grid for GridSearchCV
param_grid = {
    'C': [0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
    'gamma': ['scale', 'auto']
}
```

Nous importons le modèle SVC depuis Sklearn.svm et définissons un dictionnaire param_grid contenant les paramètres à optimiser, à savoir le paramètre de régularisation C, le type de noyau kernel et le paramètre du noyau gamma.

```
# Define the SVM model
svm = SVC()
# Perform grid search with cross-validation
grid = GridSearchCV(svm, param_grid=param_grid, scoring='accuracy', n_jobs=-1, cv=cv, verbose=1)

# Fit the model
grid.fit(X_train, y_train)
```

Nous créons un modèle SVM, effectuons une recherche sur grille avec validation croisée pour optimiser les paramètres, puis entraînons le modèle optimisé sur les données d'entraînement X_train et y_train.

```
print('Best Score:', grid.best_score_)
print('Best Params:', grid.best_params_)
print('Best Estimator:', grid.best_estimator_)
```

```
Best Score: 0.7940340909090908
Best Params: {'C': 4, 'gamma': 'scale', 'kernel': 'rbf'}
Best Estimator: SVC(C=4)
```

Le meilleur modèle SVM obtenu a un score de 0.7940 avec les paramètres optimaux C=4, gamma='scale' et kernel='rbf'.

```
# Using the best parameters from the grid-search and predicting on t

svm_grid= grid.best_estimator_
y_pred = svm_grid.predict(X_test)
```

```
svm_grid_score = accuracy_score(y_test, y_pred)
print('Model Accuracy:', svm_grid_score)
```

```
Model Accuracy: 0.8159090909090909
```

Le modèle SVM optimisé a obtenu une Accuracy de 0.8159

A Matrice de Confusion

```
# Computing the confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)

# Printing the confusion matrix
print("Confusion Matrix:")
print(conf_matrix)
```

```
Confusion Matrix:
[[164  40]
 [ 41 195]]
```

Cette matrice de confusion permet d'analyser en détail les performances du modèle SVM. Elle montre que le modèle fait à la fois de bonnes prédictions positives et négatives, avec un taux relativement faible de faux négatifs et de faux positifs.

B Évaluation des Métriques : Précision, Rappel et Score F1

```
svm_precision = precision_score(y_test, y_pred)
svm_recall = recall_score(y_test, y_pred)
svm_f1 = f1_score(y_test, y_pred)

print(f'precision: {svm_precision}')
print(f'recall: {svm_recall}')
print(f'f1: {svm_f1}')
```

```
precision: 0.8297872340425532
recall: 0.826271186440678
f1: 0.8280254777070064
```

- **La Précision :** 82,97% Le modèle a prédit correctement 83% des éléments positifs
- **Le Rappel :** 82,62% Le modèle a identifié correctement 83% des éléments positifs
- **Le Score F1 :** 82,80% Mesure équilibrée combinant précision et rappel, indiquant de bonnes performances globales

5.1.5 Decision Tree Classifier

Un arbre de décision (Decision Tree) est un modèle prédictif sous forme d'arbre qui fait des prédictions en suivant une série de décisions basées sur les caractéristiques des données.

```
from sklearn.tree import DecisionTreeClassifier
params = {
    'max_features': [1, 3, 10],
    'min_samples_split': [2, 3, 10],
    'min_samples_leaf': [1, 3, 10],
    'criterion': ["entropy", "gini"]
}
```

On a défini les paramètres à tester pour optimiser un modèle de Decision Tree Classifier de Scikit-learn, notamment le nombre maximum de caractéristiques (max_features), le nombre minimum d'échantillons requis pour diviser un nœud (min_samples_split) et pour être une feuille (min_samples_leaf), ainsi que la fonction de mesure de la qualité des divisions (criterion).

```
# Building model
dtc = DecisionTreeClassifier()

# Parameter estimating using GridSearch
grid = GridSearchCV(dtc, param_grid=params, scoring='accuracy', n_jobs =-1, cv=cv, verbose=1)

# Fitting the model
grid.fit(X_train, y_train)
```

On commence par instancier un modèle de DecisionTreeClassifier () de base.

Ensuite, on utilise GridSearchCV pour explorer systématiquement l'espace des paramètres défini précédemment dans params. Cela permet de trouver les meilleurs hyper-paramètres en maximisant le score d'Accuracy sur les données d'entraînement, en effectuant une validation croisée (cv).

Enfin, on ajuste le modèle optimisé sur les données d'entraînement (X_train, y_train) en appelant la méthode fit ().

```
print('Best Score:', grid.best_score_)
print('Best Params:', grid.best_params_)
print('Best Estimator:', grid.best_estimator_)
```

```
Best Score: 0.9068181818181819
Best Params: {'criterion': 'entropy', 'max_features': 10, 'min_samples_leaf': 1, 'min_samples_split': 2}
Best Estimator: DecisionTreeClassifier(criterion='entropy', max_features=10)
```

Le processus d'optimisation a permis de trouver un modèle de Decision Tree Classifier particulièrement performant, avec un score d'Accuracy de 90,68% sur les données d'entraînement. Les meilleurs hyper-paramètres identifiés sont un critère d'entropie pour la fonction de mesure de la qualité des divisions, 10 comme nombre maximum de caractéristiques, 1 comme nombre minimum d'échantillons dans les feuilles, et 2 comme nombre minimum d'échantillons pour diviser un nœud interne.

```
# Using the best parameters from the grid-search and predi

dtc_grid= grid.best_estimator_
y_pred = dtc_grid.predict(X_test)
```

```
dtc_grid_score = accuracy_score(y_test, y_pred)
print('Model Accuracy:', dtc_grid_score)
```

```
Model Accuracy: 0.9136363636363637
```

Après avoir optimisé les hyper-paramètres du modèle de Decision Tree Classifier via une recherche exhaustive sur la grille des paramètres, le meilleur estimateur est récupéré dans la variable dtc_grid. Ce modèle optimisé est alors utilisé pour faire des prédictions sur les données de test (X_test), donnant une précision de 91,36% sur cet ensemble de données indépendant. Ce résultat montre que le processus d'optimisation a permis d'obtenir un modèle de Decision Tree particulièrement performant, capable de généraliser efficacement ses apprentissages sur de nouvelles données.

A Matrice de Confusion

```
# Computing the confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)

# Printing the confusion matrix
print("Confusion Matrix:")
print(conf_matrix)
```

```
Confusion Matrix:
[[184  20]
 [ 18 218]]
```

La matrice de confusion montre 184 vrais négatifs, 218 vrais positifs, 20 faux positifs et 18 faux négatifs, confirmant les excellentes performances globales du modèle.

B Évaluation des Métriques : Précision, Rappel et Score F1

```
dtc_precision = precision_score(y_test, y_pred)
dtc_recall = recall_score(y_test, y_pred)
dtc_f1 = f1_score(y_test, y_pred)

print(f'precision: {dtc_precision}')
print(f'recall: {dtc_recall}')
print(f'f1: {dtc_f1}')
```

```
precision: 0.9159663865546218
recall: 0.923728813559322
f1: 0.919831223628692
```

- **La précision** de 91,60% indique que le modèle fait très peu d'erreurs de type "faux positifs", ce qui en fait un classificateur très fiable.
- **Le rappel** de 92,37% montre que le modèle a une excellente capacité à détecter les instances positives.
- **Le score F1** de 91,98% reflète une performance globale exceptionnelle, à la fois en termes de précision et de rappel.

5.1.6 Random Forest Classifier

La forêt aléatoire est un modèle très performant qui combine les prédictions de multiples arbres de décision construits de manière aléatoire

```
from sklearn.ensemble import RandomForestClassifier




# grid search cv
params={'max_depth':[3,5,10,None],

              'max_features':[1,3,5,7],
              'min_samples_leaf':[1,2,3],
              'min_samples_split':[1,2,3]
           }
```

On a un modèle de forêt aléatoire (RandomForestClassifier) issu de la bibliothèque Scikit-learn. On définit ensuite les hyper-paramètres à optimiser, comme la profondeur maximale des arbres, le nombre maximum de caractéristiques à considérer, le nombre minimum d'échantillons dans les feuilles et le nombre minimum d'échantillons pour effectuer une division.

```
# initializing random forest
rf = RandomForestClassifier()

# Parameter estimating using GridSearch
grid = GridSearchCV(rf, param_grid=params, scoring='accuracy', n_jobs =-1, cv=cv, verbose=1)

# Fitting the model
grid.fit(X_train, y_train)
```

Nous avons un modèle de forêt aléatoire (RandomForestClassifier) que nous initialisons. Ensuite, nous utilisons GridSearchCV pour trouver les meilleurs hyper-paramètres en testant différentes combinaisons de paramètres, en utilisant la validation croisée pour évaluer les performances. Finalement, nous ajustons le modèle aux données d'entraînement en utilisant la meilleure combinaison trouvée.

```
print('Best Score:', grid.best_score_)
print('Best Params:', grid.best_params_)
print('Best Estimator:', grid.best_estimator_)
```

```
Best Score: 0.9150568181818182
Best Params: {'max_depth': None, 'max_features': 7, 'min_samples_leaf': 1, 'min_samples_split': 2}
Best Estimator: RandomForestClassifier(max_features=7)
```

La recherche par grille a trouvé que le modèle de forêt aléatoire avec une profondeur maximale non limitée, 7 caractéristiques maximum par nœud, 1 échantillon minimum par feuille et 2 échantillons minimum pour diviser un nœud atteint une précision de 91,51% sur les données d'entraînement.

```
# Using the best parameters from the grid-search and pre

rf_grid= grid.best_estimator_
y_pred = rf_grid.predict(X_test)
```

```
rf_grid_score = accuracy_score(y_test, y_pred)
print('Model Accuracy:', rf_grid_score)
```

```
Model Accuracy: 0.9227272727272727
```

Le modèle optimisé par la recherche par grille fait des prédictions sur les données de test (X_test) et obtient une précision de 92,27% sur ces données.

A Matrice de Confusion

```
# Computing the confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)

# Printing the confusion matrix
print("Confusion Matrix:")
print(conf_matrix)
```

```
Confusion Matrix:
[[185  19]
 [ 15 221]]
```

B Évaluation des Métriques : Précision, Rappel et Score F1

```
rf_precision = precision_score(y_test, y_pred)
rf_recall = recall_score(y_test, y_pred)
rf_f1 = f1_score(y_test, y_pred)

print(f'precision: {rf_precision}')
print(f'recall: {rf_recall}')
print(f'f1: {rf_f1}')
```

```
precision: 0.9208333333333333
recall: 0.9364406779661016
f1: 0.9285714285714286
```

- **La précision** de 0,9208 indique que le modèle a une très bonne capacité à éviter les faux positifs.
- **Le rappel** de 0,9364 montre qu'il détecte bien les vrais positifs.
- **Le score F1** de 0,9286, le modèle a de très bonnes performances globales.

5.1.7 Artificial Neural Network

Un réseau de neurones artificiels est un modèle inspiré du cerveau humain, constitué de neurones interconnectés en couches, qui apprend à reconnaître des motifs complexes en ajustant les connexions entre les neurones.

```
 import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.callbacks import EarlyStopping
```

Le code importe les bibliothèques TensorFlow, Keras et des composants spécifiques pour construire des réseaux neuronaux, y compris les modèles séquentiels, les couches denses, les couches de dropout et la fonction EarlyStopping.

```
model = Sequential()
model.add(Dense(64, activation='relu', input_dim=X_train.shape[1]))
model.add(Dropout(0.2))
model.add(Dense(32, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Nous construisons un modèle séquentiel en ajoutant des couches denses et de dropout pour la régularisation. Les couches utilisent ReLU comme fonction d'activation, à l'exception de la couche de sortie qui utilise une activation sigmoïde. Nous compilons le modèle avec

l'optimiseur Adam, la perte d'entropie croisée binaire et la précision comme métrique.

```
# Set up early stopping
early_stopping = EarlyStopping(monitor='val_loss', patience=20, restore_best_weights=True)

# Train the model
history = model.fit(X_train, y_train, epochs=400, batch_size=32, validation_split=0.2, callbacks=[early_stopping])

# Evaluate the model on the test set
eval_result = model.evaluate(X_test, y_test)
test_accuracy = eval_result[1]
```

Nous utilisons EarlyStopping pour arrêter l'entraînement lorsque la perte de validation ne s'améliore pas pendant 20 époques. La méthode model.fit () entraîne le modèle sur X_train et y_train avec une taille de lot de 32 et une validation de 20%. Ensuite, nous évaluons le modèle sur X_test et y_test avec model. evaluate (), en extrayant la précision des résultats.

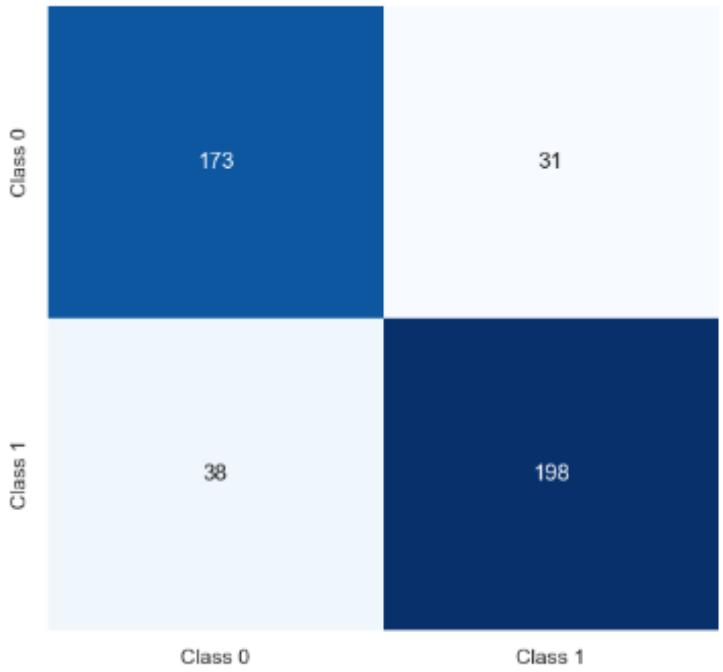
```
# Print the results
print(f'Training Accuracy: {history.history["accuracy"][-1]}')
print(f'Validation Accuracy: {history.history["val_accuracy"][-1]}')
print(f'Test Accuracy: {test_accuracy}')
```

Nous imprimons la précision d'entraînement, de validation et de test finales pour analyser les performances du modèle.

```
Epoch 155/400
44/44 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9166 - loss: 0.2152 - val_accuracy: 0.8807 - val_loss: 0.2895
Epoch 156/400
44/44 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9163 - loss: 0.1934 - val_accuracy: 0.8750 - val_loss: 0.2976
Epoch 157/400
44/44 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9085 - loss: 0.2059 - val_accuracy: 0.8636 - val_loss: 0.3059
Epoch 158/400
44/44 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9310 - loss: 0.1816 - val_accuracy: 0.8750 - val_loss: 0.2979
14/14 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8377 - loss: 0.3373
Training Accuracy: 0.9169034361839294
Validation Accuracy: 0.875
Test Accuracy: 0.8431817889213562
```

- Le modèle a été entraîné pendant 158 époques avant l'arrêt anticipé (EarlyStopping).
- Pendant l'entraînement, la précision (Accuracy) et la perte (loss) étaient suivies pour les données d'entraînement et de validation à chaque époque.
- À la 158ème époque, la précision d'entraînement était de 0.9310 et la perte de 0.1816, tandis que la précision de validation était de 0.8750 et la perte de 0.2979.
- Après l'entraînement, le modèle a été évalué sur les données de test, donnant une précision de 0.8432 et une perte de 0.3373.
- La précision finale d'entraînement était de 0.9169, la précision de validation était de 0.875, et la précision sur les données de test était de 0.8432.

A Matrice de Confusion



B Évaluation des Métriques : Précision, Rappel et Score F1

```
fnn_precision = precision_score(y_test, y_pred)
fnn_recall = recall_score(y_test, y_pred)
fnn_f1 = f1_score(y_test, y_pred)

print(f'precision: {fnn_precision}')
print(f'recall: {fnn_recall}')
print(f'f1: {fnn_f1}')
```

```
precision: 0.8646288209606987
recall: 0.8389830508474576
f1: 0.8516129032258064
```

- **La Précision :** Sur toutes les instances prédites positives, 86.46% étaient effectivement positives, indiquant un faible taux de faux positifs.
- **Le Rappel :** Sur toutes les instances réellement positives, le modèle en a correctement identifié 83.90%, indiquant un faible taux de faux négatifs.
- **Le Score F1 :** le score F1 de ce modele est de 85,16%, indiquant une réalisation globale satisfaisante.

5.2 Les modèles d'Ensemble Learning

Les modèles d'ensemble Learning combinent plusieurs modèles pour améliorer les prédictions. Des techniques comme le Boosting et le Stacking permettent d'obtenir des résultats plus fiables.

5.2.1 Boosting

A- XGBoostClassifier

XGBoostClassifier est une implémentation de l'algorithme XGBoost spécifiquement conçue pour les tâches de classification. C'est une version de l'algorithme qui est utilisée lorsque le problème implique de prédire des catégories discrètes ou des classes. XGBoostClassifier est populaire en raison de sa performance élevée, de sa capacité à gérer des ensembles de données de grande taille et de sa robustesse face à divers types de problèmes de classification.

```
from xgboost import XGBClassifier

params = {
    'max_depth': range (2, 10, 1),
    'n_estimators': range(60, 220, 40),
    'learning_rate': [0.1, 0.01, 0.05]
}
```

On a importé l'algorithme XGBClassifier de la bibliothèque XGBoost, et défini un dictionnaire de paramètres d'hyper-paramètres `params` contenant les plages de valeurs pour `max_depth`, `n_estimators` et `learning_rate`, destiné à être utilisé lors d'une recherche d'hyper-paramètres pour optimiser les performances du modèle.

```
xgb = XGBClassifier(objective='binary:logistic')

# Parameter estimating using GridSearch
grid = GridSearchCV(xgb, param_grid=params, scoring='accuracy', n_jobs =-1, cv=cv, verbose=1)

# Fitting the model
grid.fit(X_train, y_train)
```

Ensuite, on instancie XGBClassifier avec l'objectif 'binary : logistic' pour notre problème de classification binaire. On utilise GridSearchCV avec l'instance XGBClassifier, le dictionnaire params, la métrique 'Accuracy', tous les processeurs disponibles, les données de validation croisée cv, et un mode verbeux. Finalement, on entraîne le modèle avec grid.fit (X_train, y_train) pour trouver les meilleurs hyper-paramètres parmi ceux définis dans params.

```
print('Best Score:', grid.best_score_)
print('Best Params:', grid.best_params_)
print('Best Estimator:', grid.best_estimator_)
```

```
Best Score: 0.9482954545454545
Best Params: {'learning_rate': 0.1, 'max_depth': 6, 'n_estimators': 100}
Best Estimator: XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=0.1, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=6, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=100, n_jobs=None,
              num_parallel_tree=None, random_state=None, ...)
```

Nous avons obtenu un meilleur score de 0.9482en utilisant les paramètres optimaux de learning_rate à 0.1, max_depth à 6 et n_estimators à 100, avec l'estimateur XGBClassifier, ce qui démontre la robustesse et l'efficacité de notre modèle de classification après une recherche approfondie des hyper-paramètres.

```
# Using the best parameters from the grid-search and predicting

xgb_grid= grid.best_estimator_
y_pred = xgb_grid.predict(X_test)
```

```
xgb_grid_score = accuracy_score(y_test, y_pred)
print('Model Accuracy:', xgb_grid_score)
```

```
Model Accuracy: 0.95
```

Nous avons obtenu une précision de modèle de 0.95 en utilisant l'estimateur XGBClassifier optimisé (xgb_grid = grid. best_estimator_).

A Matrice de Confusion

```
conf_matrix = confusion_matrix(y_test, y_pred)

# Printing the confusion matrix
print("Confusion Matrix:")
print(conf_matrix)
```

```
Confusion Matrix:
[[188  16]
 [  6 230]]
```

B Évaluation des Métriques : Précision, Rappel et Score F1

```
xgb_precision = precision_score(y_test, y_pred)
xgb_recall = recall_score(y_test, y_pred)
xgb_f1 = f1_score(y_test, y_pred)

print(f'precision: {xgb_precision}')
print(f'recall: {xgb_recall}')
print(f'f1: {xgb_f1}')
```

```
precision: 0.9349593495934959
recall: 0.9745762711864406
f1: 0.9543568464730291
```

- **La précision** du modèle est de 93.50%, ce qui signifie que sur toutes les instances prédites comme positives, 93.50% étaient effectivement positives. Cette mesure indique un faible taux de faux positifs, ce qui est crucial dans de nombreuses applications.
- **Le rappel** du modèle est de 97.46%, ce qui indique que sur toutes les instances réellement positives, le modèle en a correctement identifié 97.46%.
- **Le score F1** du modèle est de 95.44%. Cette mesure combine à la fois la précision et le rappel, offrant ainsi une évaluation globale de la performance du modèle.

B-LightGBM

LightGBM est un modèle de Boosting d'arbres décisionnels rapide et efficace, conçu pour manipuler de grands ensembles de données et adapté aux applications nécessitant des performances en temps réel.

```
import lightgbm as lgb


# Defining all the parameters
params = {
    'max_depth': range(2, 10, 1),
    'n_estimators': range(60, 220, 40),
    'learning_rate': [0.1, 0.01, 0.05]
}
```

Nous utilisons la bibliothèque LightGBM que nous avons importée est appelée « ***lgb*** ». Ensuite, nous définissons les paramètres du modèle dans un dictionnaire appelé `params`, incluant la plage de profondeur maximale des arbres (max_depth), la plage du nombre d'estimateurs (n_estimators), et une liste de taux d'apprentissage (learning_rate).

```
# Building model
lgbm = lgb.LGBMClassifier(objective='binary')


# Parameter estimating using GridSearch
grid = GridSearchCV(lgbm, param_grid=params, scoring='accuracy', n_jobs=-1, cv=cv, verbose=1)

# Fitting the model
grid.fit(X_train, y_train)
```

Nous utilisons la classe LGBMClassifier de LightGBM pour construire notre modèle, en spécifiant l'objectif comme 'binary' pour une classification binaire. Ensuite, nous estimons les paramètres du modèle en utilisant GridSearchCV, où nous spécifions les paramètres du modèle dans param_grid, la métrique de scoring comme 'Accuracy', le nombre de tâches à exécuter en parallèle n_jobs, et utilisons une validation croisée à 10 Folds. Enfin, nous adaptons le modèle aux données d'entraînement.

```
# Best parameters found by GridSearchCV
print('Best Score:', grid.best_score_)
print('Best Params:', grid.best_params_)
print('Best Estimator:', grid.best_estimator_)
```

```
Best Score: 0.9539772727272728
Best Params: {'learning_rate': 0.1, 'max_depth': 9, 'n_estimators': 140}
Best Estimator: LGBMClassifier(max_depth=9, n_estimators=140, objective='binary')
```

Le meilleur score obtenu est de 0.9539. Les meilleurs paramètres identifiés sont : learning_rate=0.1, max_depth=9, n_estimators=140. Le meilleur estimateur obtenu est un modèle LGBMClassifier avec une profondeur maximale de 9 et 140 estimateurs, avec un objectif de classification binaire.

```
# Predicting the Labels on the test set
lgbm_grid= grid.best_estimator_
y_pred = lgbm_grid.predict(X_test)
```

```
lgbm_grid_score = accuracy_score(y_test, y_pred)
print('Model Accuracy:', lgbm_grid_score)
```

```
Model Accuracy: 0.9522727272727273
```

Nous utilisons le meilleur estimateur du modèle, issu de la recherche par grille, pour prédire les résultats sur l'ensemble de test. En calculant l'exactitude du modèle avec la fonction accuracy_score, nous obtenons un score d'exactitude de 0.9522.

A Matrice de Confusion

```
# Printing the confusion matrix
print("Confusion Matrix:")
print(onf_matrix)
```

```
Confusion Matrix:
[[189  15]
 [  6 230]]
```

Les prédictions correctes sont la somme des vrais positifs et des vrais négatifs, soit 230+189=419

Les erreurs de prédiction sont la somme des faux positifs et des faux négatifs, soit 15+6=21

B Évaluation des Métriques : Précision, Rappel et Score F1

```
lgbm_precision = precision_score(y_test, y_pred)
lgbm_recall = recall_score(y_test, y_pred)
lgbm_f1 = f1_score(y_test, y_pred)

print(f'precision: {lgbm_precision}')
print(f'recall: {lgbm_recall}')
print(f'f1: {lgbm_f1}')
```

```
precision: 0.9387755102040817
recall: 0.9745762711864406
f1: 0.9563409563409564
```

- **La Précision :** Cela signifie que sur les instances classées comme positives par le modèle, environ 93.88 % étaient réellement positives.
- **Le Rappel :** Ce score indique que le modèle a identifié environ 97.46 % des instances positives réelles.
- **Le Score F1 :** Un score F1 de 95.63 % suggère un bon équilibre entre précision et rappel.

5.2.2 Stacking

```python
def evaluate_stacking(X_train, y_train, X_test, y_test, base_models, meta_model):

    stacking_clf = StackingClassifier(
        estimators=base_models,
        final_estimator=meta_model,
        cv=10,
        n_jobs=-1
    )

    # Fit the stacking classifier
    stacking_clf.fit(X_train, y_train)

    # Evaluate the stacking classifier on the test set
    y_pred = stacking_clf.predict(X_test)
    stacking_accuracy = accuracy_score(y_test, y_pred)
    stacking_f1 = f1_score(y_test, y_pred)
    stacking_precision = precision_score(y_test, y_pred)
    stacking_recall = recall_score(y_test, y_pred)
    stacking_confusion_matrix = confusion_matrix(y_test, y_pred)

    return {
        'accuracy': stacking_accuracy,
        'f1_score': stacking_f1,
        'precision': stacking_precision,
        'recall': stacking_recall,
        'confusion_matrix': stacking_confusion_matrix,
        'model_name': type(meta_model).__name__
```

Nous avons défini une fonction evaluate Stacking qui entraîne un classificateur de Stacking et évalue ses performances sur un ensemble de test, en retournant un dictionnaire de métriques.

```
# Define the base models
base_models = [
    ('xgb', XGBClassifier()),
    ('lgbm', LGBMClassifier())
]

# Define the meta-models to test
meta_models = [
    LogisticRegression(),
    RandomForestClassifier(),
    SVC(),
    XGBClassifier(),
    DecisionTreeClassifier(),
    GaussianNB(),
    KNeighborsClassifier(),
    LGBMClassifier()
]
```

Ensuite on a défini deux modèles de base et plusieurs méta-modèles pour évaluer leurs performances dans un classificateur de Stacking.

```
# Evaluate the stacking classifier with different meta-models
for meta_model in meta_models:
    metrics = evaluate_stacking(X_train, y_train, X_test, y_test, base_models, meta_model)
    print(f"Stacking Classifier Metrics with {metrics['model_name']}:")
    print(f"Accuracy: {metrics['accuracy']:.4f}")
    print(f"F1-score: {metrics['f1_score']:.4f}")
    print(f"Precision: {metrics['precision']:.4f}")
    print(f"Recall: {metrics['recall']:.4f}")
    print("Confusion Matrix:")
    print(metrics['confusion_matrix'])

    # Display the confusion matrix
    disp = ConfusionMatrixDisplay(confusion_matrix=metrics['confusion_matrix'])
    disp.plot()
    plt.title(f"Confusion Matrix for {metrics['model_name']}")
    plt.show()
    print()

    # Update best model metrics if current model is better
    if best_metrics is None or metrics['accuracy'] > best_metrics['accuracy']:
        best_metrics = metrics
```

On a évalué chaque méta-modèle en utilisant la fonction définie, affiché les métriques et la matrice de confusion, puis mis à jour les meilleures métriques si le modèle actuel était meilleur.

```
# Print the best model's metrics
print("#### BEST MODEL ####")
print(f"Best Model: {best_metrics['model_name']}")
print(f"Accuracy: {best_metrics['accuracy']:.4f}")
print(f"F1-score: {best_metrics['f1_score']:.4f}")
print(f"Precision: {best_metrics['precision']:.4f}")
print(f"Recall: {best_metrics['recall']:.4f}")
print("Confusion Matrix:")
print(best_metrics['confusion_matrix'])

# Display the confusion matrix of the best model
disp = ConfusionMatrixDisplay(confusion_matrix=best_metrics['confusion_matrix'])
disp.plot()
plt.title(f"Confusion Matrix for Best Model: {best_metrics['model_name']}")
plt.show()
```

Nous avons affiché les métriques et la matrice de confusion du meilleur modèle après avoir évalué tous les méta-modèles.

```
#### BEST MODEL ####
Best Model: SVC
Accuracy: 0.9614
F1-score: 0.9648
Precision: 0.9433
Recall: 0.9873
Confusion Matrix:
[[190  14]
 [  3 233]]
```

Le meilleur modèle trouvé parmi tous les méta-modèles testés est le Support Vector Classifier avec Accuracy de 96.14%.

6. Discussions et résultats

Comme illustré dans la section précédente, dans notre démarche de proposition de modèle performant répondant adéquatement au problème d'irrigation automatique, nous avons commencé à tester et à observer le comportement des modèles Machine Learning de base après leur apprentissage avec les données du Dataset utilisé. Nous avons constaté un avantage significatif des modèles Random Forest et Decision Tree par rapport aux modèles logistic Regression, Naive baise, SVM, Artificial neural network et KNN sur toutes les mesures.
L'avantage moyen peut s'élever à ***25.75%*** entre les modèles de la même classe. Le meilleur modèle de cette gamme, qui est Random Forest, a donné un avantage moyen de ***1.01%*** par

rapport à son concurrent qui est DT, avec les performances suivantes Random Forest Classifier ***Accuracy : 91.14%, Precision : 91.21%, Recall : 92.37% et F1-Score : 91.78%***.

Un résumé des résultats de ces modèles est présenté dans la première partie du **Tableau IV.1**. Dans un deuxième temps, nous avons procédé aux tests sur les modèles de la classe d'Ensemble Learning, plus précisément XGBoost et LightGBM. Le premier modèle XGBoost a donné un avantage moyen de ***4.23%*** par rapport au meilleur modèle de la classe précédente Random Forest. Nous avons pu améliorer cet avantage à ***4.49%*** sur toutes les mesures en utilisant LightGBM. Les performances de ce modèle s'élèvent à ***95,22 %*** pour ***l'Accuracy, Precision de 93.87%, Recall 97.45% et F1-Score de 95.63%.***

Pour aller plus loin dans notre démarche d'optimisation des performances, nous avons implémenté un modèle de Stacking, utilisant LightGBM et XGBoost comme base learners, et Support Vector Machine comme Meta Learner. Cette approche vise à combiner les forces des deux meilleurs modèles de la classe d'Ensemble Learning avec la capacité du SVM à trouver des marges optimales pour séparer les classes. Les performances du modèle de Stacking sont les suivantes : ***Accuracy : 96.14%, Precision : 94.33%, Recall : 98.73% et F1-Score : 96.48%.***

Ces résultats montrent une amélioration notable par rapport aux performances du modèle Random Forest, soulignant l'efficacité de la stratégie de Stacking dans notre contexte. Le modèle de Stacking offre une amélioration de ***5.49%*** en termes d'Accuracy, ***5.12%*** en termes de F1-Score, ***3.42%*** en termes de Precision, et ***6.89%*** en termes de Recall par rapport à au meilleur modèle de la classe précédente Random Forest.

Tableau IV.1 Les algorithmes utilisé

Modèle	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Logistic Regression	0.7250	0.7385	0.7542	0.7463
K-Nearest Neighbors	0.7386	0.7249	0.8262	0.7722
Gaussian Naïve Bayes	0.7727	0.7982	0.7711	0.7844
Support Vector Machines	0.8159	0.8297	0.8262	0.8280
Artificial Neural Networks	0.8431	0.8711	0.8305	0.8503
Decision Tree Classifier	0.9022	0.8906	0.9322	0.9109
RANDOM FOREST Classifier	0.9113	0.9121	0.9237	0.9178
XGBoost	**0.9500**	**0.9349**	**0.9745**	**0.9543**
Light_GBM	**0.9522**	**0.9387**	**0.9745**	**0.9563**
Staking	**0. 9614**	**0.9433**	**0.9873**	**0.9648**

Après avoir choisi notre meilleur modèle, nous l'avons déployé dans une interface web locale L'utilisateur entre les données pertinentes dans l'interface présentée dans Figure IV.6 et cliquer sur la Button 'Get Prediction Result' L'interface affiche la prédiction du modèle, indiquant si le robinet doit être ouvert ou non.

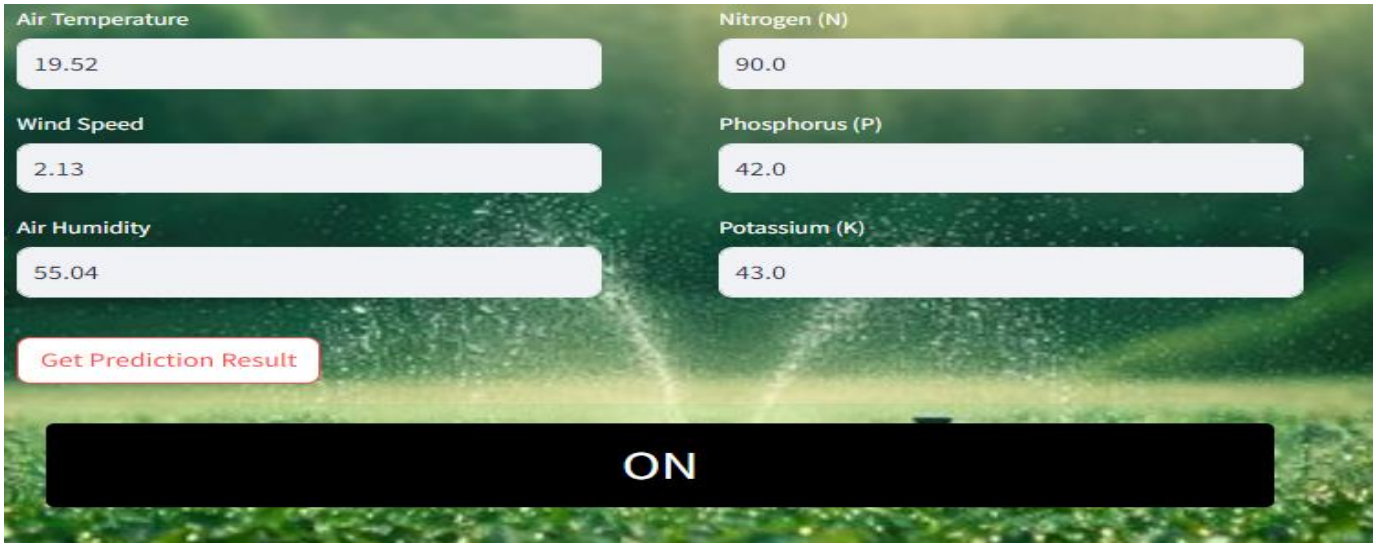


Figure IV.6 Interface web locale

Conclusion

Dans ce chapitre, nous avons exploré le développement et la mise en œuvre d'un système d'irrigation intelligent visant à optimiser l'utilisation de l'eau dans le secteur agricole. Nous avons commencé par une description détaillée du système d'irrigation étudié, en soulignant l'importance de chaque composant, des capteurs à l'interface utilisateur.

Le processus de mise en place de la solution d'irrigation a été décrit en plusieurs étapes clés, allant du déploiement des capteurs à l'entraînement des modèles d'apprentissage automatique, en passant par l'évaluation et l'intégration du système. Nous avons également présenté l'environnement de travail utilisé pour notre étude, en détaillant les équipements matériels et les bibliothèques logicielles employées.

L'une des parties centrales de notre étude a été la mise en œuvre et l'évaluation de différents modèles de Machine Learning. Nous avons testé une variété de modèles, des techniques

d'apprentissage de base comme Logistic Regression et KNN, jusqu'aux méthodes d'Ensemble Learning telles que le Boosting et le Stacking.

En résumé, ce travail démontre le potentiel des technologies intelligentes pour améliorer l'efficacité de l'irrigation agricole. L'intégration de capteurs avancés, de modèles de Machine Learning et d'interfaces utilisateur intuitives permet de créer un système d'irrigation adaptatif et précis. L'intégration des techniques d'Ensemble Learning, et particulièrement du modèle de Stacking, nous a permis d'atteindre des performances optimales pour notre problème d'irrigation automatique. Ces résultats confirment l'intérêt de combiner différents modèles pour tirer parti de leurs complémentarités et améliorer la robustesse des prédictions.

Conclusion générale

Conclusion générale

En conclusion, ce mémoire met en lumière le potentiel des technologies intelligentes pour transformer l'agriculture moderne. L'irrigation de précision, rendue possible par l'intégration de capteurs avancés, de modèles de Machine Learning, et d'interfaces utilisateur intuitives, représente une avancée significative vers une gestion agricole plus efficiente et durable.

Grace à la démarche suivie au cours de ce travail, nous avons pu mettre en place un modèle performant en utilisant la méthode de Stacking d'Ensemble Learning pour résoudre le problème d'irrigation automatique. Cette approche nous a permis d'améliorer significativement les performances par rapport aux modèles de base et aux autres modèles d'Ensemble Learning testés. En combinant les forces des modèles LightGBM et XGBoost avec le Support Vector Machine, nous avons obtenu des résultats supérieurs en termes de précision par rapport aux résultats obtenus jusqu'à maintenant sur le même jeu de données. Ces résultats prometteurs confirment la robustesse des techniques employées et attestent de leur efficacité.

Ce travail constitue un point de départ prometteur pour d'autres recherches dans ce domaine, démontrant le potentiel de l'Ensemble Learning pour résoudre des problèmes complexes et ouvrant la voie à de futures optimisations et applications. L'application du Machine Learning dans l'agriculture de précision offre des perspectives prometteuses en essayant d'optimiser l'irrigation, la fertilisation et la gestion des cultures grâce à l'analyse de grandes quantités de données. Notre projet peut être étendu, en proposant des modèles qui peuvent prévoir les besoins en eau, détecter précocement les maladies et recommander les moments optimaux pour la récolte, améliorant ainsi l'efficacité, la productivité et réduisant les coûts et l'impact environnemental.

Bibliographie

Bibliographie

[1] Bricout, M., Roussel, R., & Monteil, C. (2022). « Agriculture de précision : Définition …». INRAE. Dictionnaire d'agroécologie. https://doi.org/10.17180/vjvj-2g81.hal-03648446.

[2] Kogut, P. (1970). « Agriculture de précision : L'Industrie du futur » . EOS Data Analytics. https://eos.com/fr/blog/agriculture-de-precision/ (Dernier accès le 27 mai 2024)

[3] Li, M., & Chung, S. (2015). "Special issue on precision agriculture. Computers and Electronics in Agriculture", 112, 1. 2024. https://doi.org/10.1016/j.compag.2015.03.014

[4] Routledge eBooks. (2013). "Precision Agriculture for sustainability and environmental Protection". pp 3 - 20 https://doi.org/10.4324/9780203128329

[5] Abobatta, W. F. (2022). "Why we need precision agriculture?" Journal of Applied Biotechnology & Bioengineering. Dernier accès le 27 mai 2024

[6] Demin, A. (s.d.). « Les avantages et les inconvénients de l'agriculture de précision selon les agriculteurs néerlandais ». BE PROFY IN POTATOES NEWS.. https://fr.potatoes.news/the-pros-and-cons-of-precision-agriculture-according-to-dutch-arable-farmers/ Dernier accès le 27 mai 2024

[7] Shah, F., & Wu, W. (2019). "Soil and Crop Management Strategies to Ensure Higher Crop Productivity within Sustainable Environments". Sustainability, 11(5), 1485. Dernier accès le 27 mai 2024.

[8] Tang, Y., Chen, C., Leite, A. C., & Xiong, Y. (2023). "Precision control technology and application in agricultural pest and disease control". Frontiers in Plant Science. Dernier accès le 27 mai 2024.

[9] Sharma, A., Jain, A., Gupta, P., & Chowdary, V. (s.d.). "Machine Learning Applications for Precision Agriculture: A Comprehensive Review". Dernier accès le 27 mai 2024.

[10] Nicole. (2021). « Precision Irrigation - Hong Kong. Hong Kong ». https://itrade.gov.il/hongkong/2021/10/25/precision-irrigation/. Dernier accès le 27 mai 2024

[11] Robert, P. C., Rust, R. H., Larson, W. E., & Lange, A. F. (1996). "Centimeter Accuracy Differential GPS for Precision Agriculture Applications. Precision Agriculture ».. doi: 10.2134/1996.precisionagproc3.c81 .

[12] Dealer Spike, "Precision Farming RTK Plus" | Birkey's Farm Store | Champaign. Birkey's Farm Store Champaign, All Rights Reserved. https://www.birkeys.com/--precision-farming-rtk-plus Dernier accès le 27 mai 2024

[13] Shanwad, U. K., Patil, V. C., Dasog, G. S., Mansur, C. P., & Shashidhar, K. C. (s.d.). "Global Positioning System (GPS) in Precision Agriculture". University of Agricultural Sciences Dharwad. Dernier accès le 27 mai 2024.

[14] Ahirwar, S., Swarnkar, R., Bhukya, S., & Namwade, G. (s.d.). 'Application of Drone in Agriculture'. College of Agricultural Engineering and Technology, Anand Agricultural University. Dernier accès le 27 mai 2024.

[15] Satellites, drones, capteurs, robots. . ., les nouveaux outils de l'agriculture. (n.d.). INRAE Institutionnel. https://www.inrae.fr/actualites/dossier-teledetections Dernier accès le 27 mai 2024.

[16] Veroustraete, F. (2015). The rise of the drones in agriculture. EC agriculture. Dernier accès le 27 mai 2024 .

[17] Palomino, W., Morales, G., Huaman, S., & Telles, J. (2018). PETEFA: Geographic Information System for Precision Agriculture. IEEE XXV International Conference on Electronics, Electrical Engineering and Computing (INTERCON). Dernier accès le 27 mai 2024.

[18] Hammonds, T. (2023). 'Use of GIS in agriculture - Cornell Small Farms. Cornell Small Farms'. Dernier accès le 27 mai 2024, https://smallfarms.cornell.edu/2017/04/use-of-gis/

[19] Components of a GIS. (n.d.). Dernier accès le 27 mai 2024 https://www.rst2.org/ties/GENTOOLS/comp_gis.html

[20] Spore Magazine. (2017). L'agriculture de précision : une approche intelligente et sur mesure. Dernier accès le 27 mai 2024.

[21] Stuart Russell et Peter Norvig, "ArtificialIntelligence: A Modern Approach", 3ème édition.

[22] Machine Learning - Introduction et premier algorithme - CodiMD. (s. d.). Consulté le 20 mai 2024, à l'adresse https://markdown.data-ensta.fr/p/machine-learning-introduction?print-pdf#/

[23] Domingos, P. (2015). "The master algorithm: How the quest for the ultimate learning machine will remake our world". Basic Books.

[24] Belaidi, N. (n.d.-b). L'apprentissage supervisé : définition et exemples. Formation Tech et Data en ligne | Blent.ai. Consulté le 15 avril 2024, à l'adresse https://blent.ai/blog/a/apprentissage-supervise-definition

[25] Jvc, J. (2022). *Introduction au Machine Learning avec Python*. Data Transition Numérique. . Consulté le 23 avril 2024, à l'adresse https://www.data-transitionnumerique.com/machine-learning-python/

[26] Wikiwand - Binary classification. (n.d.). Wikiwand. Consulté le 15 avril 2024, à l'adresse https://www.wikiwand.com/en/Binary_classification

[27] Brownlee, J. (2020, août 19). 4 Types of Classification Tasks in Machine Learning. MachineLearningMastery.com. Consulté le 21 avril 2024, à l'adresse https://machinelearningmastery.com/types-of-classification-in-machine-learning/

[28] Subha. (2024, 24 janvier). Machine Learning Basics : K Nearest neighbors - subha - medium. *Medium*. Consulté le 15 avril 2024, à l'adresse

https://medium.com/@pingsubhak/machine-learning-basics-k-nearest-neighbors-9e8e2d46db75

[29] Jasmin PrafulBharadiya. (Volume. 8 Issue. 5, May - 2023) "Role of Teachers as a Leader to developEnvironmentalAwarenessamongStudents for a Sustainable World." International Journal of Innovative Science and ResearchTechnology (IJISRT), www.ijisrt.com. ISSN - 2456-2165, PP :- 2504-2508. https://doi.org/10.5281/zenodo.8020795

[30] Wikiwand - Régression logistique. (2011, 1 novembre). Wikiwand. Consulté le 23 avril 2024, à l'adresse https://www.wikiwand.com/fr/R%C3%A9gression_logistique

[31] Breiman, L. (2001). Random Forests. Machine Learning, 45, 5–32. https://doi.org/10.1023/A:1010933404324

[32] BigdataMa, T. (2020, February 9). *La régression linéaire*. LE BIGDATA Au Maroc Et L'Intelligence Artificielle Au MAROC. Consulté le 23 avril 2024, à l'adresse https://www.bigdata.ma/la-regression-lineaire

[33] Chapitre V : Régression. (n.d.). Introduction À L'apprentissage Automatique. Consulté le 4 mai 2024, à l'adresse https://projeduc.github.io/intro_apprentissage_automatique/regression.html

[34] Kassel, R. (2023, 9 novembre). Apprentissage Non Supervisé : principe et utilisation. Formation Data Science | DataScientest.com. Consulté le 30 avril 2024, à l'adresse https://datascientest.com/apprentissage-non-supervise

[35] Chapelle, O., Schölkopf, B., &Zien, A. (2006). A Discussion of Semi-Supervised Learning and Transduction. In O. Chapelle, B. Schölkopf, & A. Zien (Eds.), Semi-Supervised Learning (pp. 473-478). Cambridge, MA, USA: MIT Press.

[36] Sutton, R. S., &Barto, A. G. (2018). Reinforcementlearning: An introduction. MIT press.

[37] Robert, J. (2023, November 9). *Reinforcement Learning : Définition et application*. Formation Data Science | DataScientest.com.Consulté le 30 avril 2024, à l'adressehttps://datascientest.com/reinforcement-learning

[38] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

[39] Haykin, S. S. (2009). Neural networks and learning machines. Pearson Education.

[40] Kassel, R. (2024 , mars 14). Fine-Tuning : Qu&rsquo ; est-ce que c&rsquo ; est ? À quoi ça sert en IA ? Formation Data Science | DataScientest.com. Consulté le 18 mai 2024, à l'adresse https://datascientest.com/fine-tuning-tout-savoir#:~:text=Le%20fine%2Dtuning%20est%20une,Learning%20sur%20une%20t%C3%A2che%20sp%C3%A9cifique.

[41] BotPenguin. (2024, January 16). Fine-tuning: process, best practices & pitfalls| BotPenguin. Consulté le 18 mai 2024, à l'adresse https://botpenguin.com/glossary/fine-tuning

[42] Hesaraki, S. (2023, 15 novembre). Transfer learning - Saba Hesaraki - Medium. Medium. Consulté le 20 mai 2024 à l'adresse https://medium.com/@saba99/transfer-learning-bbf6b67deb88

[43] S.J. Pan et Q. Yang (2010) "A Survey on Transfer Learning"

[44] Dahiya, R. (2024, mai 19). Federated Learning: training AI models while preserving data privacy. Medium. Consulté le 20 mai 2024, à l'adresse https://medium.com/@rijuldahiya/federated-learning-training-ai-models-while-preserving-data-privacy-ef1fdef64f55

[45] Contributeurs aux projets Wikimedia. (2024, 7 mai). Apprentissagefédéré. Consulté le 20 mai 2024, à l'adresse https://fr.wikipedia.org/wiki/Apprentissage_f%C3%A9d%C3%A9r%C3%A9

[46] Li, T., Sahu, A., Talwalkar, A., & Smith, V. (2019). Federated Learning: challenges, methods, and future directions. ResearchGate.

[47] Seni, G., & Elder, J. F. (2010). Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions.

[48] Opitz, D., & Maclin, R. (1999). Popular Ensemble Methods: An Empirical Study.

[49] Tariverdiyev, N. (2021, December 7). Machine Learning Algorithms: Ensemble Methods, Bagging, Boosting and Random Forests. Medium. Consulté le 20 mai 2024 https://medium.com/@nadir.tariverdiyev/machine-learning-algorithms-ensemble-methods-bagging-boosting-and-random-forests-7d3df7adfab8

[50] Schapire, R. E. (2003). The boosting approach to machine learning: An overview.

[51] Robert, J. (2023, November 9). *Algorithmes de boosting – AdaBoost, Gradient Boosting, XGBoost*. Formation Data Science | DataScientest.com. Consulté le 20 mai 2024 https://datascientest.com/algorithmes-de-boosting-adaboost-gradient-boosting-xgboost

[52] Wikiwand - LightGBM. (n.d.). Wikiwand. Consulté le 20 mai 2024 https://www.wikiwand.com/en/LightGBM

[53] GeeksforGeeks. (2023, May 23). Boosting in Machine Learning: Boosting and AdaBoost. GeeksforGeeks Consulté le 20 mai 2024 https://www.geeksforgeeks.org/boosting-in-machine-learning-boosting-and-adaboost/

[54] Zhou, Z. (2012). Ensemble Methods: Foundations and Algorithms.

[55] Pimple, J. F., Sharma, A., & Mishra, J. K. (2024). MediSecure: A Blockchain-Enabled Ensemble Learning Approach for User-Controlled Single Sign-On and Privacy Preservation in Medical Cyber-Physical Systems. In Ponnusamy, S., & Bora, V. R. (Eds.), The Role of AI in Bio-Medical Translations' Research for the Health Care Industry. AIBTR 2023. Communications in Computer and Information Science. Springer, Cham. https://doi.org/10.1007/978-3-031-49454-3_5

[56] G. Sambasivam and G. D. Opiyo,
A predictive Machine Learning application in agriculture: Cassava disease detection and classi cation
with imbalanced dataset using convolutional neural networks," Egyptian
Informat. J., early access, doi: 10.1016/j.eij.2020.02.007.

[57] J. Amara, B. Bouaziz, and A. Algergawy, A deep learning-based
approach for banana leaf diseases classi cation," in Datenbanksysteme
für Business, Technologie und Web (BTW 2017) Workshopband,
B. Mitschang, D. Nicklas, F. Leymann, H. Schöning, M. Herschel,
J. Teubner, T. Härder, O. Kopp, and M. Wieland, Eds. Bonn, Germany:
Gesellschaft für Informatik e.V.

[58] T. Rumpf, A.-K. Mahlein, U. Steiner, E.-C. Oerke, H.-W. Dehne, and
L. Plümer, Early detection and classi cation of plant diseases with
support vector machines based on hyperspectral re ectance,"
.

[59] B. Espejo-Garcia, N. Mylonas, L. Athanasakos, S. Fountas, and
I. Vasilakoglou, Towards weeds identi cation assistance through
transfer learning,"

[60] T. J. Glezakos, G. Moschopoulou, T. A. Tsiligiridis, S. Kintzios, and
C. P. Yialouris, ``Plant virus identi cation based on neural networks with
evolutionary preprocessing,"

[61] E. Acar, M. S. Ozerdem, and B. B. Ustundag, ``Machine learning based
regression model for prediction of soil surface humidity over moderately
vegetated elds," in Proc. 8th Int. Conf. Agro-Geoinformat. (Agro-
Geoinformat.).

[62] S. Park, J. Im, S. Park, and J. Rhee, ``AMSR2 soil moisture downscaling using multisensor products through machine learning approach," in Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS).

[63] R. Reda, T. Saffaj, B. Ilham, O. Saidi, K. Issam, L. Brahim, and E. M. El Hadrami, ``Acomparative study between a newmethod and other machine learning algorithms for soil organic carbon and total nitrogen prediction using near infrared spectroscopy," Chemometric Intell. Lab. Syst.

[64] J. Diez-Sierra and M. D. Jesus, ``Long-term rainfall prediction using atmospheric synoptic patterns in semi-arid climates with statistical and machine learning methods," J. Hydrol.

[65] K. Mohammadi, S. Shamshirband, S. Motamedi, D. Petkovi¢, R. Hashim,and M. Gocic, ``Extreme learning machine based prediction of daily dew point temperature," .

[66] Y. Cai, K. Guan, D. Lobell, A. B. Potgieter, S. Wang, J. Peng, T. Xu, S. Asseng, Y. Zhang, L. You, and B. Peng, integrating satellite and climate data to predict wheat yield in Australia using machine learning approaches," Agricult. Forest Meteorol.

[67] S. Kulkarni, S. N. Mandal, G. S. Sharma, M. R. Mundada, and Meeradevi, Predictive analysis to improve crop yield using a neural network model," in Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI), Bangalore, India.

[68] P. Nevavuori, N. Narra, and T. Lipping, Crop yield prediction with deep convolutional neural networks," Comput. Electron. Agricult.

[69] S. A. Gyamerah, P. Ngare, and D. Ikpe, Probabilistic forecasting of crop yields via quantile random forest and Epanechnikov kernel function," Agricult. Forest Meteorol.

[70] S. Shahinfar and L. Kahn, ``Machine learning approaches for early prediction of adult wool growth and quality in Australian merino sheep," Comput. Electron. Agricult.

[71] Abioye, E.A.; Hensel, O.; Esau, T.J.; Elijah, O.; Abidin, M.S.Z.; Ayobami, A.S.; Yerima, O.; Nasirahmadi, A. Precision Irrigation Management Using Machine Learning and Digital Farming Solutions. AgriEngineering. https://doi.org/10.3390/ agriengineering4010006

[72] Glória, A.; Cardoso, J.; Sebastião, P. Sustainable Irrigation System for Farming Supported by Machine Learning and Real-Time Sensor Data. Sensors 2021, 21, 3079. https://doi.org/10.3390/s21093079

[73] Mohammed MAA, Kaya F, Mohamed A, Alarifi SS, Abdelrady A, Keshavarzi A, Szabó NP and Szűcs P (2023), Application of GIS-based machine learning algorithms for prediction of irrigational groundwater quality indices
doi: 10.3389/feart.2023.1274142. Front. Earth Sci. 11:1274142.

[74] Vallejo-Gómez, D.;Osorio, M.; Hincapié, C.A. Smart Irrigation Systems in Agriculture: A Systematic Review. https://doi.org/10.3390/ agronomy13020342

[75] Proposed smart irrigation system architecture. (n.d.). ResearchGathttps://www.researchgate.net/figure/Proposed-smart-irrigation-system-architecture_fig3_343776611 Consulté le 30 mai 2024

[76] Suite de capteurs sans fil pour Vantage Pro 2 - 6322OV - Davis Instruments. (n.d.). Météo Shopping. https://www.meteo-shopping.com/fr/suite-de-capteurs/97-suite-de-capteurs-sans-fil-pour-vantage-pro-2.html Consulté le 30 mai 2024

[77] OLED 0,91" et GY-BME280 sur ESP8266 D1 Mini. (n.d.). AZ-Delivery. https://www.az-delivery.de/fr/products/oled-bme280-d1mini-bundle Consulté le 30 mai 2024

[78] Goy, M. (2015, May 6). Parrot Flower Power. Les Numériques. https://www.lesnumeriques.com/objet-connecte/parrot-flower-power-p23905/test.html` Consulté le 30 mai 2024

[79] Dataset for Predicting watering the plants. (2021). https://www.kaggle.com/datasets/nelakurthisudheer/dataset-for-predicting-watering-the-plants Consulté le 30 mai 2024