Water World

Adrien Chabert

Travail de Bachelor

Juin 2019

Table des matières

Introduction

L'eau, H_2O, est à première vue une ressource abondante sur notre terre. En effet, plus de 70% de la surface de la terre est recouvert d'eau. On la retrouve sous forme liquide (mers, lacs, rivières, nappes), sous forme de glace et sous forme de gaz (dans les nuages, dans l'atmosphère). Cependant, seulement 3% de cette eau est de l'eau douce, soit 35 millions de kilomètres cubes. A titre de comparaison, le lac Léman a un volume de 89km^3.

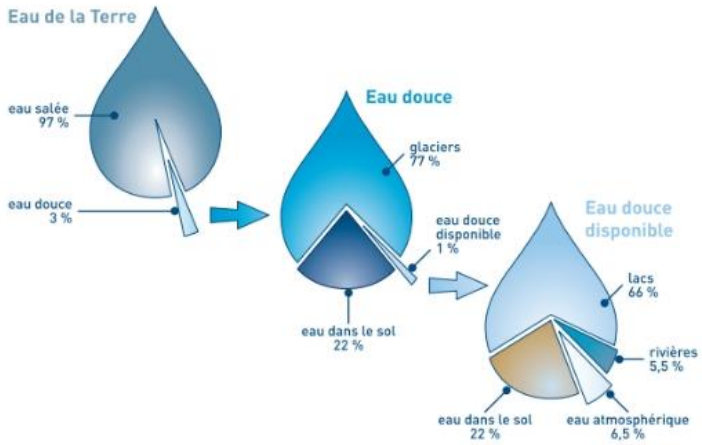

Figure 1 : Eau sur Terre. Source : « L'eau de la Terre : origine et répartition ». Onglet Pédagogique 7e Continent *(blog).*

Sur ces 3% d'eau douce 77% se trouve sous forme de glace et 22% est profondément enfoui sous Terre. Ainsi sur l'ensemble de l'eau douce sur terre, seulement 1% est liquide et disponible que ce soit sous forme de lac, de rivière, dans les nappes phréatiques ou dans l'atmosphère. De plus, la Russie, l'Inde, le Canada, les États-Unis, l'Indonésie, le Congo et la Chine se partagent à eux neufs 60% du débit mondial d'eau. L'or bleu est donc inégalement réparti.

L'eau douce est essentielle à la vie sur Terre pour tous les êtres vivants. Pour l'Homme, elle est utilisée pour s'hydrater, pour l'agriculture, pour les usages ménagers ou encore pour l'industrie. La consommation mondiale d'eau douce en 2000 était 4 km^3 par an (1,3 millions de litres) par an et par habitant. Le besoin d'eau est en hausse car il y a de plus en plus d'être humain.

70% de notre consommation en eau est utilisé par l'agriculture. En considérant que de nombreux pays se trouvent déjà en stress hydrique une grande partie de l'année et que ce nombre ne tend pas à décroître, améliorer l'agriculture, en la rendant moins consommatrice d'eau, est un des enjeux majeurs de l'humanité pour les prochaines années.

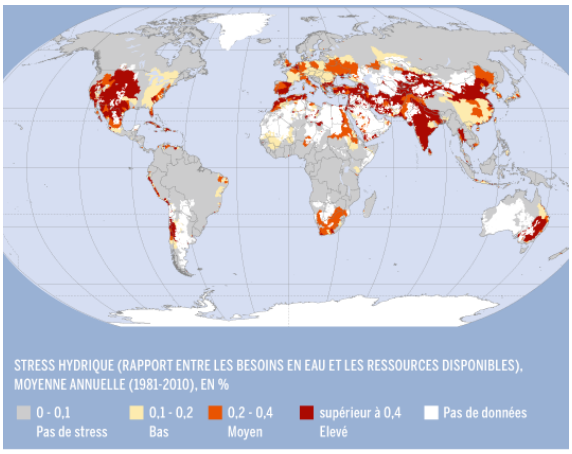

Figure 2 : Carte du monde du stress hydrique. (Valo, 2015).

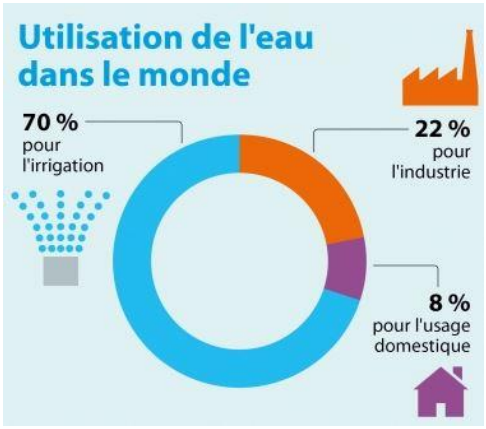

Figure 3 : Utilisation de l'eau. (« Planetoscope - Statistiques : Consommation d'eau dans le monde », 2019).

Différentes méthodes ont déjà été étudiées : cultiver des espèces qui nécessitent une moins grande quantité d'eau ou encore améliorer les systèmes d'irrigation. C'est ce dernier point qui est traité dans ce projet et plus précisément : comment utiliser l'eau de manière plus efficiente. Irriguer les champs avec la quantité d'eau minimum nécessaire au bien-être des plantations et suivre un planning d'arrosage efficace permettraient d'éviter un surplus d'arrosage et donc un gaspillage d'eau.

Ainsi le but de mon projet est de prédire la quantité d'eau nécessaire pour garder la terre à un certain taux humidité sur une période donnée. Tout mon projet se trouve dans le GitHub suivant : https://github.com/cui-unige/BSc-sensor-machine-learning.

Installation

Une importante phase du projet est la phase de récolte de données. Pour ce faire, deux bacs à plantes ont été mis à disposition avec, pour chacun, un arrosage automatique journalier programmable et des capteurs sensorielles. Les bacs se trouvent dans des pièces fermées à proximité de fenêtre.

Un premier bac a été placé dans une salle bien-exposée, orientée sud-ouest. Ce bac sera nommé *Demeter*. Dans celui-ci est planté des oignons et du basilic.

Le deuxième est placé dans une salle moins bien exposée, orientée nord-est et sera nommée *Cérès*. Dans ce bac, des épinards sont cultivés.

Ces emplacements ont été choisis car les épinards demandent une plus faible exposition que le basilic et les oignons. Dans les bacs, un arrosage goutte-à-goutte pour chaque pied de plantes a été installé. Les capteurs sont placés à environ deux centimètres de la zone d'arrosage du goutte-à-goutte.

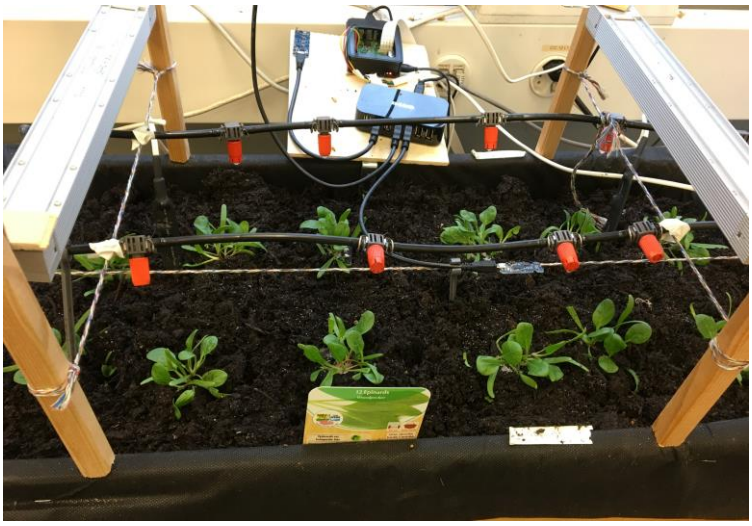

Figure 4 : Photo du bac de Cérès. On peut voir des plants d'épinard, un arrosage goutte-à-goutte avec des goûteurs en orange et des capteurs plantés dans la terre à gauche et à droite.

Chaque bac dispose de deux capteurs enterrés et d'un capteur de luminosité indépendant. Ces capteurs fournissent de très nombreuses informations utiles au calcul de l'évolution de l'humidité dans le sol. Les deux capteurs enterrés permettent de récolter des informations sur l'humidité, la température et l'ensoleillement. Les données sont récoltées toutes les 30 secondes.

En ce qui concerne l'arrosage, des pompes permettent le transfert d'eau d'un réservoir aux goutte-à-goutte. Ces dernières s'activent à 11h pendant l'heure d'hiver et à midi pendant l'heure d'été. Il nous ait possible de définir le temps de pompage. C'est pourquoi l'arrosage ne sera pas mesuré en litre mais en seconde, qui correspondent au temps que la pompe transfert de l'eau aux goutteurs.

Humidité du sol

Le pourcentage d'humidité dans le sol est le facteur principal de ce projet. Le but de ce projet étant de prédire la quantité d'eau nécessaire pour garder un certain taux d'humidité après une durée déterminée. Ce taux est calculé en pourcentage. Il s'agit du rapport entre la masse d'eau présente dans le sol et la masse du sol. Mettre le capteur dans de l'eau reviendrait à obtenir une humidité de 100%.

Facteurs impactant l'humidité

L'humidité dans le sol est influencée par différents facteurs. Selon les études, ces facteurs sont :

- L'ensoleillement,
- La quantité d'eau arrosée,
- Le vent,
- Le type de terre,
- La qualité de la terre et des végétaux-

Toutefois, plusieurs de ces facteurs n'ont pas été pris en compte. Premièrement, le vent, même s'il joue un grand rôle à l'extérieur (en asséchant les terres, n'influe nullement puisque les bacs ont été placés à l'intérieur d'une salle fermée. Deuxièmement, le type de terre n'est pas significatif car la même terre a été utilisée dans les deux bacs. En effet, il s'agit de terre universelle que l'on peut trouver communément dans le commerce. Troisièmement, l'ensoleillement n'influe pas car les capteurs ont été rapidement protégés du soleil par le feuillage des plantes et car l'ensoleillement est fortement corrélé à la température.

Enfin, l'évolution de la plante n'a également pas été considérée car celle-ci est très difficile à calculer et les résultats, qui seront exposés par la suite, montrent le faible impact de ce facteur sur l'humidité.

La température, l'humidité du sol actuel et l'évolution de la terre ont été pris en compte pour la prédiction du taux d'humidité dans le sol. L'analyse de ces facteurs est expliquée dans la section « *Choix des facteurs à fournir à notre apprentissage »* du chapitre « *Méthodologie ».*

Taux d'humidité optimal

Afin d'utiliser l'eau de manière plus efficace, il faut déterminer le taux d'humidité optimal à l'évolution de la plante. Celui-ci dépend du type de plante et du type de sol. Je n'ai trouvé aucun chiffre définissant le taux d'humidité optimal du sol pour une plante. Mais, il est connu que le besoin en eau des oignons est relativement restreint alors que celui du basilic et des épinards est important. Ainsi l'humidité du sol pour la culture d'oignons doit être en général inférieure à celle du basilic et des épinards.

En ce qui concerne l'humidité suivant le type de sol, une plus grande documentation est disponible (Rebecca Shortt, Anne Verhallen, & Pam Fisher, 2019). Chaque type de sol a des caractéristiques différentes en fonction de sa consistance. La capacité de rétention et le point de flétrissement permanent sont des caractéristiques essentielles à la détermination de l'humidité d'un sol.

La capacité de rétention correspond à la quantité maximale d'eau qu'un sol peut contenir. Ce taux peut être calculer en quantifiant la quantité d'eau deux à trois jours après un arrosage saturant le sol. En dessus de ce taux, il est inutile d'arroser car le sol est incapable d'absorber de l'eau supplémentaire et donc, qui engendrera du gaspillage d'eau qui est à éviter absolument dans le cadre d'une optimalisation de l'utilisation de l'eau.

Le point de flétrissement permanent correspond à l'humidité du sol à laquelle une plante, à l'aide de ses racines, n'est plus capable d'en extraire l'eau. Le sol est trop sec. En dessous de ce point, le sol est incapable de subvenir aux besoins des plantes. Entre ces deux taux (la capacité de rétention et le point de flétrissement) se trouve la quantité d'eau utilisable. Plus cette plage d'eau est grande, plus un sol est propice au développement de plantes. Il ne faut pas attendre le point de flétrissement pour arroser car plus on est proche de ce seuil, plus la plante aura des difficultés à s'approvisionner en eau. Pour l'arrosage par goutte-à-goutte, il est conseillé d'arroser lorsqu'il ne reste plus que 80% d'eau disponible. Alors que pour l'irrigation par aspersion, il est conseillé d'irriguer à partir de 50% d'eau disponible.

Le sol utilisé pour notre expérience est une terre universelle qui peut facilement être acheté dans le commerce. Ceci correspond à un sol de type de loam limoneux. Ainsi le but de notre logiciel sera de rester au-dessus des 25% d'humidité dans notre sol. La plage d'eau disponible est d'environ 15%.

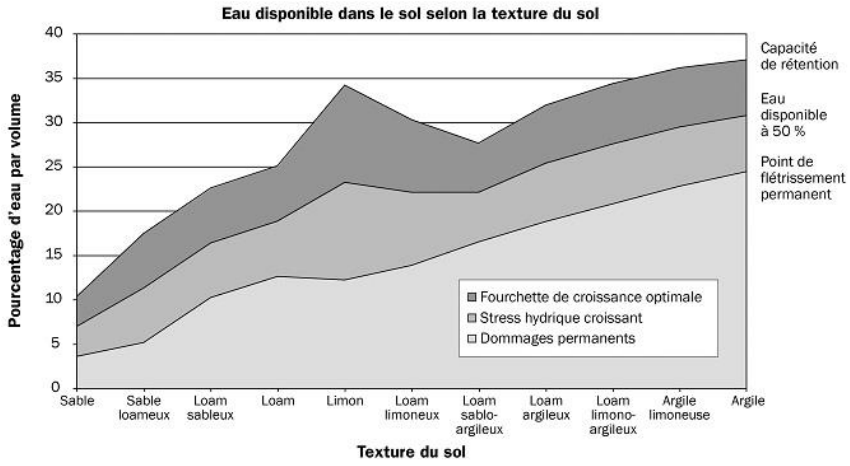

Figure 5 : Humidité optimal pour un sol. (Shortt, Verhallen, & Fisher, 2019).

Documentation

ConservWater

Une large étude a été mené par l'institut de technologie de Californie à Pasedena et cette dernière a publié un article approfondi dans le IJOEAR (Journal international de recherche en environnement et en agriculture) nommé « *Global custom-tailored machine learning of soil water content for locale specific irrigation modeling with high accuracy* » écrit par Aadith Moorthy en 2016. Cet article fait une analyse de la situation actuelle en matière d'arrosage et un comparatif entre différents algorithmes de Machine Learning pour l'arrosage. Il propose un algorithme de Machine Learning dédié à la prédiction d'arrosage pour la culture de plants dans le monde entier. Ils ont créé un algorithme utilisant du Machine Learning. Cet algorithme est nommé ConserWater™ algorithm.

Selon leurs analyses, ils ont montré que leur algorithme se comporte mieux que des algorithmes statiques tel que Aqua Crop, proposé par Land and Water Division of the Food and Agricultural Organization (FAO). Leur algorithme obtient également de meilleurs résultats que des algorithmes de Machine Learning basé sur les itérations dans le temps tels que ARIMA et SARIMA. Ceci vient du fait que ARIMA et SARIMA ont besoin de données avec des propriétés statiques. Ce qui ne peut pas être le cas, étant donné que la météo ne suit aucune loi statiques (déviation standard ou moyenne).

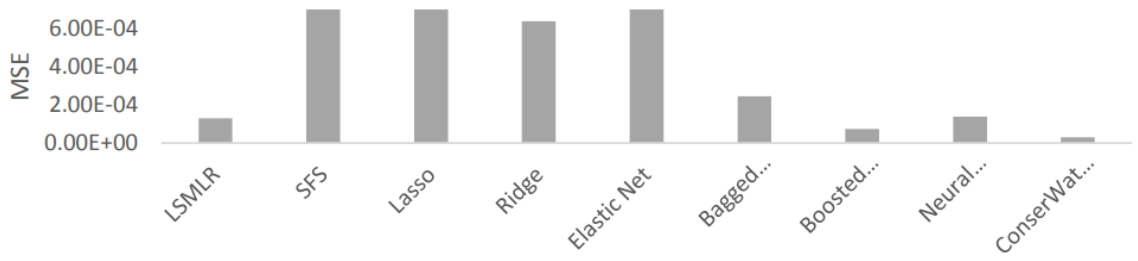

Figure 6: Erreur moyenne au carré (MSE) sur 30 jours pour différents algorithmes en ne donnant que l'humidité de la première journée. (Moorthy & Aadith, 2016).

On peut voir que leur algorithme (ConserWater) obtient un bien meilleur résultat que d'autres algorithmes de machine Learning classique et plus fréquemment utilisé. Il est intéressant de noter que la régression linéaire multiple avec la méthode des moindres carrés (LSMLR) obtient de très bon résultat par rapport aux autres algorithmes de Machine Learning. Ceci est d'autant plus intéressant que c'est un algorithme avec une faible complexité d'implémentation.

A noter qu'il est important de séparer les données par rapport à leur région. En effet l'humidité ne réagit pas de façon identique dans une région aride que dans une région tropicale.

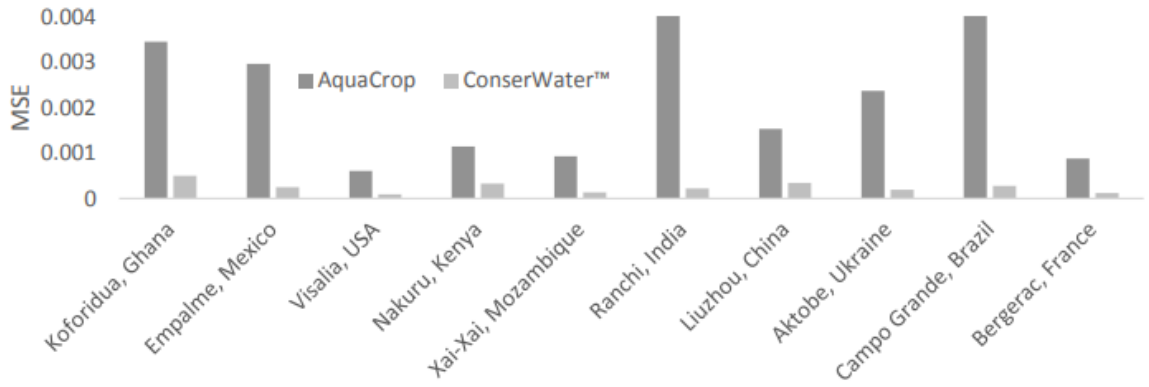

Figure 7: Erreur MSE entre Aqua Crop et ConserWater suivant les régions. (Moorthy & Aadith, 2016).

L'article affirme que l'utilisation de Machine Learning peut permettre une économie d'eau de 25% par rapport à l'usage habituel, mais que l'utilisation de capteurs sensoriels installés par les maraichers peut revenir à un coût rédhibitoire pour les petites fermes. C'est pourquoi ils ont développé un algorithme utilisant les données récoltées par les nombreux capteurs météorologiques déjà existants, notamment en utilisant le *World Weather online service*. Ils ont à disposition 30'000 stations de collectes de données qui s'étalent sur un peu plus d'une année.

Une telle étude n'est pas du tout comparable à celle menée dans ce projet. Le nombre de données collectées est amplement supérieur aux nôtres. De plus cette étude californienne a été menée en extérieur et sur différents continents. Ainsi, le nombre de facteur impactant l'humidité est beaucoup plus important que pour notre étude en intérieur. Il leur a fallu prendre en compte le vent, les averses, la condensation, la rosée, etc. Cependant il est intéressant de noter qu'une simple régression linéaire permet d'obtenir des résultats très satisfaisants et que l'économie possiblement réalisée est conséquente.

Autres études

Estimating soil moisture using remote sensing data: A machine learning approach

Cette étude (Ahmad, Sajjad, Ajay Kalra, & Haroon Stephen) réalisée en 2009 sur des données récoltées au Colorado a duré 5 ans. Il s'agit d'estimer l'humidité du sol avec les données du capteur en utilisant la technique appelée Support Vector Machine (SVM). Ils ont obtenu de meilleurs résultats avec la SVM qu'avec les modèles de réseau de neurones et qu'avec la régression linéaire à variables multiples. Cependant cette méthode ne sera pas utilisée dans ce projet car elle est très complexe.

Planning d'arrosage hebdomadaire avec Jojoba Israel

Jojoba est une société qui vend des capteurs d'humidité. L'entreprise a réalisé une étude (Goldstein et al., 2010) sur des données récoltées à l'aide de 22 capteurs sur 2 ans. Le but de leur expérience était de faire un planning hebdomadaire d'arrosage. Ils ont comparé différents algorithmes d'apprentissage et ont conclu que le meilleur résultat était obtenu avec Gradient Boosted Regression Trees. En effet le taux de précision est de 93%. Ils ont jugé leurs résultats satisfaisants et ont estimé que leurs résultats aidaient les ingénieur agronome à mieux définir leur arrosage.

Machine Learning

Les algorithmes de Machine Learning ou autrement appelé algorithme d'apprentissage sont des algorithmes qui permettent à la machine d'apprendre à partir de données en se basant sur les statistiques de celles-ci. On parle alors d'intelligence artificielle. Le plus connu de ces algorithmes est Alpha GO développé par DeepMind qui a permis de battre au jeu de Go des professionnels. Avant l'arrivée d'Alpha GO, le jeu de Go était considéré comme impossible à simuler pour un logiciel tant son nombre de possibilité de jeu est immense.

Le principe d'un algorithme d'apprentissage est le suivant. L'algorithme d'apprentissage reçoit une large base de données. Ces données contiennent des valeurs pour différents paramètres et un résultat pour cette donnée, on parle dans ce cas d'apprentissage supervisé. Puis à partir de cette base de données, l'algorithme construit un modèle qui lui permettra de prédire un résultat en fonction des paramètres utilisés pour l'apprentissage (paramètres qui lui ont été fournis dans la base de données). Dans le cas où il n'y a pas de résultats fournis pour chaque donnée de la base de données, on parle alors d'apprentissage non supervisé. L'algorithme ne fournit pas une prédiction, mais est alors capable de construire des structures sous-jacentes aux données. Par exemple, si l'on fournit une large base de données sur des animaux, l'algorithme non-supervisé peut être capable de les catégoriser : mammifères, amphibiens, poissons, oiseaux, etc.

Ainsi, pour mon travail, il est indispensable d'utiliser un apprentissage supervisé étant donné que l'on veut prédire la quantité d'humidité du sol. Il existe deux classes d'apprentissage supervisé différentes, comme la classification et la régression. Lorsque les résultats sont discrets, par exemple le résultat est une espèce d'iris, on parle alors de classification. Lorsque les résultats sont continus, on utilise des régressions. J'ai utilisé ces dernières car la valeur de l'humidité est continue entre 0 et 100 (on parle de pourcentage d'humidité).

Régression

Régression linéaire multiples méthode des moindres carrés

Il s'agit du modèle classique de régression linéaire qui est le plus facilement utilisable. Il consiste à utiliser un vecteur w = (w_0, w_1, …, w_p) qui servira de coefficient pour prédire notre résultat.

$$\hat{y}_i = w_0 + w_1 * x_{i,1} + w_2 * x_{i,2} + \cdots + w_p * x_{i,p}$$

Il faut trouver le meilleur vecteur w qui minimise l'erreur selon la formule des moindres carrés entre la prédiction ($\hat{y}$) et le résultat escompté (y). Plus il y a de dépendance entre les facteurs, plus la méthode des moindres carrés devient très sensible. C'est pourquoi il faut faire attention aux données que l'on utilise. Cette méthode n'est pas conseillée, s'il n'y a aucune vérification des données.

$$\min_w \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Régression Ridge

Deux caractéristiques sont importantes à l'estimation de la qualité de prédiction.

La première est le biais. Il s'agit du carré de l'écart en l'espérance de la prédiction et la valeur réelle.

La deuxième est la variance ou l'instabilité, c'est-à-dire, la dispersion de la prédiction. Afin de minimiser cette instabilité on ajoute une contrainte sur les coefficients. Il s'agit d'une

contrainte L2 sur la taille du coefficient (voir la formule ci-dessous). Plus l'alpha est élevé, plus les coefficients sont robustes. La régression linéaire par la méthode des moindres carrés ordinaires peut conduire à un système d'équations surdéterminé ou sous-déterminé.

$$\min_w \sum_{i=1}^{n} (y_i - \sum_{j=0}^{p} w_j * x_{ji})^2 + \alpha * \sum_{j=0}^{p} {w_j}^2$$

Régression de Lasso

Cette régression est à utiliser lorsque l'on souhaite réduire le nombre de variables dont dépend la solution donnée. Il faut effectuer une régularisation L1 sur les coefficients w. Ainsi, on tend vers le principe du rasoir d'Ockham, faire un système aussi simple que possible, mais pas forcément la meilleure. S'il existe une forte corrélation entre les composants, Lasso privilégiera l'un au détriment de l'autre ce qui peut nuire à la qualité du modèle.

$$\min_w \sum_{i=1}^{n} (y_i - \sum_{j=0}^{p} w_j * x_{ji})^2 + \alpha * \sum_{j=0}^{p} w_j$$

Elasticnet

Il s'agit d'une combinaison entre la méthode Ridge et la méthode Lasso. Il faut combiner une régression L1 et une régression L2 sur les coefficients w. Elasticnet permet de tirer les avantages des deux méthodes, c'est-à-dire la capacité de sélection des variables et d'exclusion des variables non pertinentes (Lasso) ainsi que le partage des poids des variables corrélées (Ridge).

$$\min_w \sum_{i=1}^{n} (y_i - \sum_{j=0}^{p} w_j * x_{ji})^2 + \alpha * \sum_{j=0}^{p} {w_j}^2 + \alpha * \sum_{j=0}^{p} w_j$$

ARIMA

ARIMA (Autoregression Models for Time Series Forecasting) est un modèle de prédiction pour les séries temporelles. Cet algorithme est conçu pour les systèmes où la valeur à prédire dépend d'observations effectuées à des périodes précédentes. Concrètement une régression linéaire est effectuée à partir des observations précédentes.

Cette méthode est utilisée pour des données qui font preuve de propriété stochastique non statistique[1]. Ceci n'est pas le cas de notre projet étant donné que la météo ne suit aucune loi. Elle est aléatoire et indépendante du temps.

Gradient Boosting

Le concept du boosting est le suivant. Un modèle de prédiction est construit et des prédictions sont établies. Les résultats des prédictions sont classifiés en fonction de leur difficulté à prédire, c'est-à-dire, la réussite de la prédiction. Un poids faible est affecté aux données dites

[1] Un processus est statique ou stationnaire quand ses propriétés qui le caractérise sont indépendante du temps.

« de difficulté facile » et un poids fort est attribué aux données dites « à forte difficulté ». Ensuite un nouveau modèle est créé en prenant en compte la difficulté des données dans le but d'améliorer la prédiction. Ce schéma est répété jusqu'à obtenir une prédiction correcte. Ce système permet d'identifier les faiblesses d'apprentissage et d'en améliorer celui-ci.

Le Gradient Boosting est légèrement différent. Ce dernier ne cherche pas à minimiser les pertes mais à optimiser une fonction fournie par l'utilisateur. Ce changement correspond mieux à l'usage du monde réel.

Méthodologie

Afin de créer un système capable de conseiller l'arrosage sur une période donnée, il faut lui fournir un algorithme de prédiction. Pour ce faire, un algorithme d'apprentissage doit lui être fourni ainsi que des données qui permettront à celui-ci d'appendre. Plusieurs choix sont nécessaires à la réussite de l'apprentissage.

- Comment récolte-t-on les données ?
- Quelles données doivent être fourni à l'algorithme et celle qui ne doivent pas ?
- Quel algorithme d'apprentissage utiliser ?
- Comment améliorer cet algorithme ?

Récolte de données

Je rappellerai ici que les deux bacs ne sont pas exposés pareillement, ne contiennent pas les mêmes plantes et les capteurs ne sont pas tous placés à la même distance d'un arrosage.

Dans chacun de ces deux bacs se trouvent deux capteurs d'humidité, de température et de luminosité. Chaque valeur des capteurs est saisie toutes les 15 secondes. Les données ont été récoltées sur trois mois, de mi-mars à mi-juin.

L'arrosage se fait sur un point spécifique, étant donné que c'est un type d'arrosage goutte à goutte. Ainsi, l'humidité du sol varie en fonction de la distance du capteur à un point d'arrosage. Les deux capteurs dans le bac Cérès (bac avec plants à épinards) et un des capteurs dans Demeter (bac avec plants à oignons et à basilic) sont à une distance de 2 centimètres d'un point d'arrosage. Alors que le deuxième capteur dans Demeter est à une distance de 4 centimètres. C'est la raison pour laquelle j'ai choisi de ne pas utiliser ce dernier capteur.

Les épinards, qui ont été plantés dans le bac Cérès, sont arrivés à maturité deux mois après la plantation, c'est-à-dire mi-mai, et ce sont détériorés par la suite. Alors que les basilics et oignons étaient toujours en bonne santé mi-juin. Ces différentes se manifestent sur les courbes d'humidité des capteurs de leur bac respectif. Ci-dessous un graphique de l'erreur[2] si on rassemble des deux bacs ou si on les sépare. C'est pourquoi j'ai décidé de ne pas rassembler les données des deux bacs et d'appliquer des algorithmes d'apprentissage indépendamment entre les bacs.

[2] L'erreur utilisée sur tous mes calculs est la suivante : elle correspond à la moyenne de la différence en valeur absolue entre la prédiction et la valeur réelle pour chaque itération (toutes les 30 minutes)

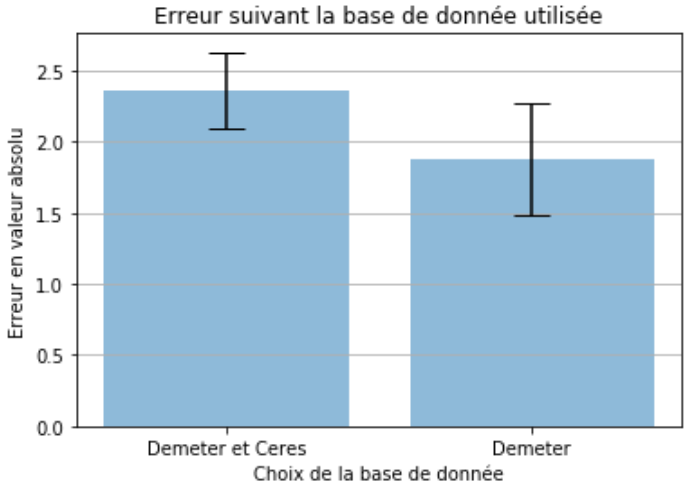

Figure 8 : Erreur en valeur absolu en fonction du choix des données utilisés.

Il est inutile de récupérer les données toutes les 15 secondes car les capteurs sont parfois imprécis et comportent du bruit. Il est préférable de récolter les données toutes les 30 minutes afin de récupérer une moyenne des facteurs sur 30 minutes. Ceci permet d'éviter l'apparition de bruit sur les valeurs obtenues. Chaque facteur suit ainsi une courbe homogène.

L'arrosage a été journalier et a toujours été programmé à 11h pour l'heure d'hiver et à midi pour l'heure d'été. Ainsi la représentation graphique de chaque journée comporte un pic à 11h ou à 12h, puis une phase d'évaporation rapide de l'eau et enfin une troisième phase de stabilisation de l'humidité. A noter que les deuxième et troisième phases sont beaucoup plus marquées pour Demeter que Cérès.

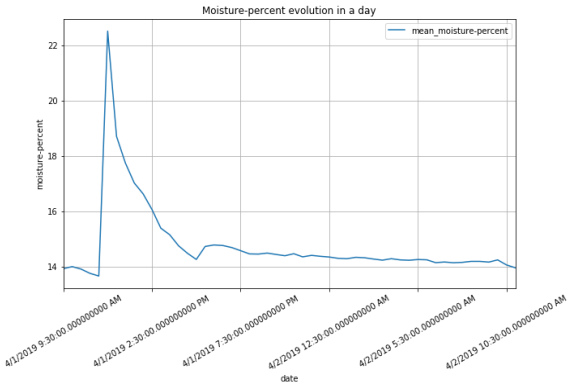

Figure 9 : Evolution classique de l'humidité dans le bac Demeter. L'arrosage pour cette journée était de 20 secondes à midi.

Toutes les journées sont séparées. Une journée commence à l'arrosage et se termine à l'arrosage suivant. Ainsi chaque journée, qui contient 48 itérations (une itération toutes les 30 minutes), est utilisée pour l'apprentissage de l'algorithme de prédiction.

Il est important d'avoir un échantillon donné d'apprentissage qui représente toutes les combinaisons d'arrosage. C'est pourquoi il faut varier la quantité d'arrosage. Un arrosage de 10, 15, 20, 35, 40 et 45 secondes[3] a été effectué pour le bac Demeter et un arrosage de 10, 15, 20 et 30 secondes a été effectué pour le bac Cérès. Il est important de noter que x secondes pour Déméter ne correspond pas à x secondes pour Cérès. En effet x secondes de pompage

[3]L'arrosage se mesure en secondes. Cette quantité représente le temps de pompage de l'eau qui sera ensuite déverser aux plantes par l'intermédiaire d'un arrosage goûte à goûte.

d'eau chez Demeter doit approvisionner en eau 10 goutte-à-goutte. Alors x secondes de pompages d'eau chez Cérès doit approvisionner 6 goutte-à-goutte. Ainsi, à seconde équivalente, un arrosage chez Cérès est deux fois plus conséquent que pour Déméter.

Élimination des données inappropriées

Des problèmes de récolte de données sont apparus pendant les 3 mois de récolte, comme notamment un capteur défectueux ou un arrosage incomplet ou encore une panne avec le Raspberry pi qui collecte les données. C'est pourquoi il a fallu vérifier la qualité des données et supprimer celles qui étaient incomplètes ou incorrectes. Chaque journée qui comporte des itérations incomplètes ont été supprimées. Un autre moyen aurait pu être choisi, telle qu'une interpolation linéaire des données manquantes. Cependant, il est arrivé que des données manquaient sur plusieurs journées. Ainsi, il était impossible d'interpoler correctement les données manquantes sans induire en erreur notre apprentissage. Les journées avec des arrosages doubles ou incomplets ont également été supprimées. Celles-ci influençaient en erreur notre algorithme d'apprentissage.

Lors de la mise en place du système, une nouvelle terre a été utilisée pour l'expérience. La terre était donc relativement humide et ne réagissait pas de la même façon à un arrosage qu'un mois après l'installation de celle-ci. C'est pourquoi les données des quatre premières journées ont également été supprimées, car la terre n'était pas encore suffisamment stabilisée.

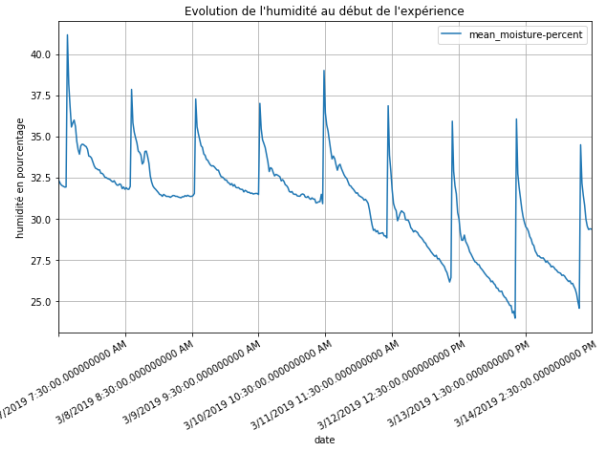

Figure 10 : Evolution de l'humidité durant les 8 premiers jours de l'expérience. Chaque jour a été arrosé identiquement mais ne réagit pas de la même façon.

Choix des facteurs à fournir à notre apprentissage

Comme mentionné précédemment, l'humidité du sol est influencée par les précipitations, par la température, la luminosité et le vent.

Dans cette expérience, il s'agit d'une étude en intérieur, c'est pourquoi le facteur du vent est exclu. La luminosité n'est également pas prise en compte, car elle est fortement corrélée par la température et nos bacs ont une très forte densité de feuillage. Ainsi, les rayons du soleil ne touchent pas directement le sol et nos capteurs. De plus, la seule précipitation perçue par les bacs à plantes est l'arrosage automatique programmé quotidiennement. Ainsi, les facteurs restants sont la température, l'humidité de base dans le sol et l'arrosage.

Comme on peut le voir à la figure 9, la variation de l'évolution de l'humidité dans le sol n'est pas régulière. Pendant les quatre premières heures après l'arrosage, une forte baisse de l'humidité est constatée. Ensuite, la baisse d'humidité est beaucoup plus faible. C'est pour cela que j'ai jugé bon de prendre comme facteur le temps depuis le dernier arrosage et la

quantité de ce dernier (*ArrosageHist)*. Important, *ArrosageHist* et *Arrosage* ne sont pas les mêmes valeurs. *Arrosage* indique la valeur de l'arrosage à une date et une heure précise. Ainsi quand on n'arrose pas, la valeur de cet arrosage est de 0. Alors que *ArrosageHist* indique la valeur du dernier arrosage effectué. Ainsi pendant toutes une journée la valeur reste inchangée.

Un dernier facteur peut être également pris en compte : il s'agit du nombre de journée écoulées depuis le début de l'expérience (*Index)*. En effet, on peut imaginer que la terre, qui était à l'origine neuve, s'appauvrisse et donc qu'elle retienne moins facilement l'eau et par conséquent l'humidité. Les plantes ont également crû pendant l'expérience et donc on peut supposer logiquement que les plantes pompent plus d'eau à la fin plutôt qu'au début l'expérience.

Pour rassembler toutes ces valeurs, j'ai utilisé la librairie pandas dans python qui permet d'utiliser des dataframes. Les dataframes sont très pratiques pour la manipulation de données. Elles s'apparentent à des matrices où les colonnes peuvent être nommées. Voici une représentation de dataframe que j'ai utilisé pendant mon travail. Elle contient tous les paramètres cités auparavant.

	date	mean_moisture-percent	mean_temperature	moistureAdd	temperatureAdd	Arrosage	TAfterArrosage	ArrosageHist	index
150	3/10/2019 10:30:00.000000000 AM	31.476667	24.918333	5.533333	0.161667	10	0	10	4
151	3/10/2019 11:00:00.000000000 AM	37.010000	25.080000	-1.550000	0.186667	0	30	10	4
152	3/10/2019 11:30:00.000000000 AM	35.460000	25.266667	-0.663333	0.968333	0	60	10	4
153	3/10/2019 12:00:00.000000000 PM	34.796667	26.235000	-0.223333	-0.628333	0	90	10	4
154	3/10/2019 12:30:00.000000000 PM	34.573333	25.606667	-0.250000	-0.285000	0	120	10	4
155	3/10/2019 1:00:00.000000000 PM	34.323333	25.321667	-0.440000	0.306667	0	150	10	4
156	3/10/2019 1:30:00.000000000 PM	33.883333	25.628333	-0.423333	1.411667	0	180	10	4

Figure 11 : Représentation d'une dataframe avec les informations sur la date, l'humidité, la température, l'arrosage et la journée de collecte de données.

Afin de déterminer l'importance de ces différents facteurs, un comparatif de l'erreur obtenu en fonction des paramètres utilisés a été établi. Pour mener ce test, j'ai choisi de prendre une régression linéaire multiple à moindre carré. J'ai fait ce choix, car dans mes recherches, j'ai pu voir que la LSMLR permettait d'obtenir de très bons résultats. Le graphique suivant montre l'erreur obtenue en variant les facteurs pris en compte. D'après mes recherches et mes résultats, la température, l'humidité de base, la quantité d'arrosage et le temps après arrosage sont des facteurs primordiaux pour une bonne précision de prédiction. Selon mes observations, le temps depuis le début de l'expérience et l'historique de l'arrosage ne permettent pas d'améliorer les performances de l'algorithme, ainsi ils ne sont pas pris en compte.

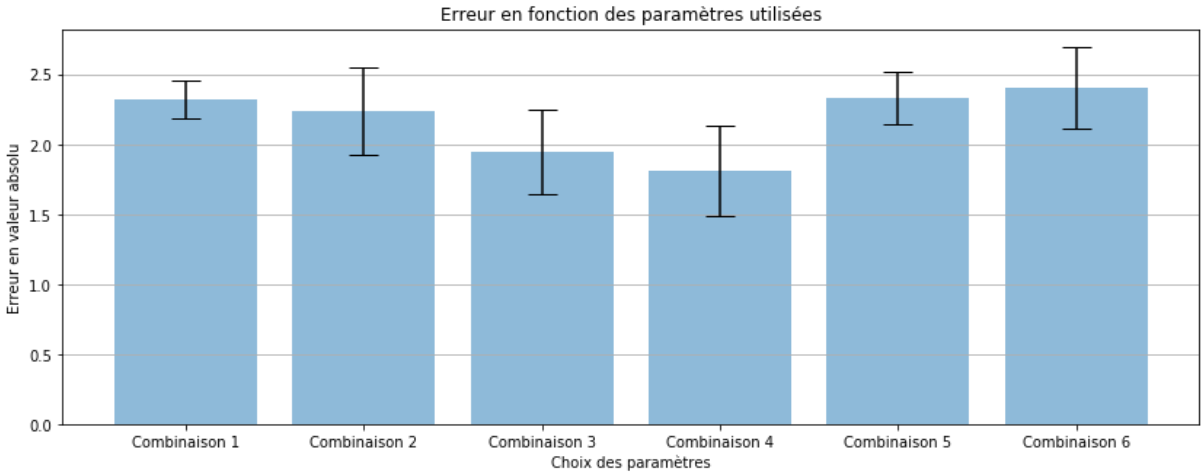

Figure 12 : Graphique de l'erreur en fonction des paramètres utilisés pour la régression linéaire multiple. Combinaison 1 : l'arrosage et l'humidité actuelle. Combinaison 2 : l'arrosage, l'humidité et la température actuelle. Combinaison 3 : l'arrosage et l'humidité actuelle et le temps depuis le dernier arrosage. Combinaison 4 : l'arrosage, l'humidité et la température actuelle et le temps depuis le dernier arrosage. Combinaison 5 : l'arrosage et l'humidité actuelle et la valeur du dernier réel arrosage effectué. Combinaison 6 : l'arrosage et humidité actuelle et combien de journées se sont écoulées depuis le début de l'expérience.

Choix de l'algorithme d'apprentissage

Afin de trouver le meilleur algorithme pour prédire l'humidité sur une période donnée et ainsi créer un programme d'arrosage précis, j'ai effectué un comparatif entre plusieurs algorithmes. Le but étant de faire la meilleure prédiction d'humidité sur une journée entière. La prédiction doit être aussi proche que possible à n'importe quel moment de la journée. Pour rappel, une journée commence à l'arrosage et se termine à l'arrosage suivant. Une seule régression était faite sur toute la journée avec les paramètres sélectionnés ci-dessous.

Pour mener à bien des tests, il faut séparer les données en deux : une partie de données d'apprentissage (données de training) et une autre partie de données de test. Il ne faut pas utiliser les mêmes données pour apprendre que pour tester. De plus, cette séparation doit être totalement aléatoire. Les données utilisées pour le choix de l'algorithme sont celles fournies par le bac Demeter. Après élimination des journées erronées ou incomplètes, le nombre de journées restantes est de 91. 77 journées ont été utilisées pour apprendre et 14 pour tester, ce qui fait un ratio d'environ 84% de données d'apprentissage et 16% de données de test.

Le calcul de l'erreur se fait de la manière suivante :il s'agit de la moyenne de la différence en valeur absolue entre la prédiction et la valeur réelle pour chaque itération.

$$erreur = \frac{\sum_{i=1}^{nombre\ d'itération} |prédicton(i) - valeur_{réelle}(i)|}{nombre\ d'itération}$$

Grâce à mes recherches j'ai choisi de tester les algorithmes suivants :

- Régression linéaire multiple moindre carré
- ElasticNet
- Gradient Boosting Regression.

Pour la régression Elasticnet, il est nécessaire de fournir un paramètre d'apprentissage. Il s'agit du taux de régularisation. Ainsi il est nécessaire de trouver ce meilleur taux. Il m'a fallu varier ce taux et en garder le meilleur.

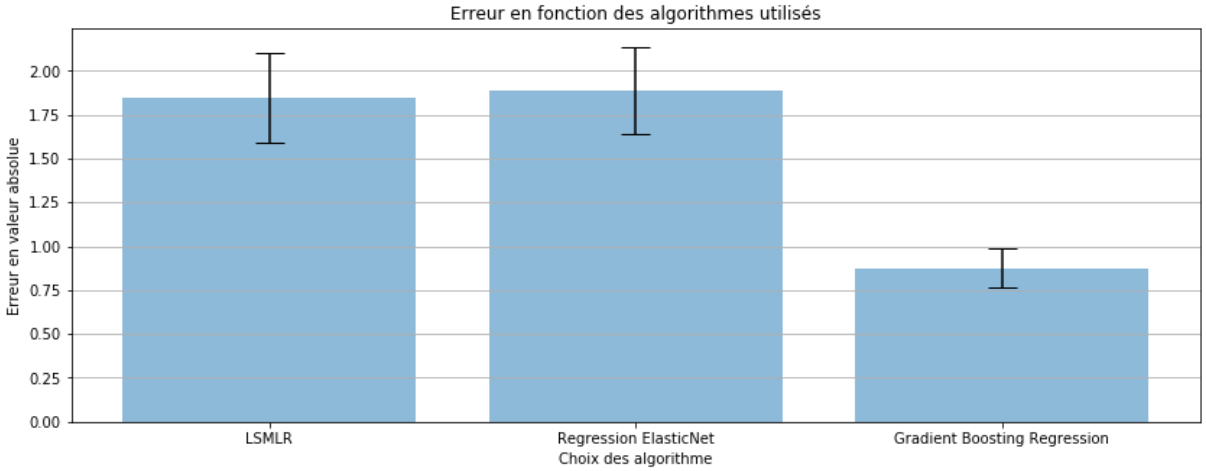

Figure 13 : Moyenne d'erreur en fonction de l'algorithme utilisé.

On peut voir que la régression Gradient Boosting permet d'obtenir de très bons résultats. Dans le chapitre suivant, l'algorithme LSMLR sera fortement améliorer jusqu'à atteindre une erreur similaire à la régression Gradient Boosting.

Amélioration de l'algorithme

Comme mentionné dans la partie *Récolte de données* (page 10), une journée peut être séparée en 3 parties distinctes. La première, très courte (30 minutes), correspond à l'arrosage. La deuxième à la période d'évaporation ou infiltration de l'eau arrosé. Enfin la troisième correspond à la partie de stabilisation de l'humidité avec une légère perte d'humidité. Cette dernière perdure jusqu'au prochain arrosage.

Afin d'améliorer la précision, chaque journée est séparée suivant ces trois parties et une régression linéaire est appliquée sur chacune de ces trois parties distinctement. Ceci permet grandement d'améliorer la précision de prédiction. La première partie est uniquement les 30 minutes qui suivent l'arrosage. La deuxième et troisième partie se partagent le reste de la journée. La séparation a été définie de sorte qu'elle minimise l'erreur.

Est-ce que cette limite dépend de la quantité de précipitation ?

On peut penser que plus la précipitation est conséquente, plus la période d'évaporation sera longue. Ainsi pour chaque quantité d'arrosage fourni, j'ai cherché la meilleure séparation. Voici des graphiques représentant l'erreur en fonction de la variation pour chacun des arrosages effectués dans l'ordre chronologique (10, 20, 40, 35, 45, 15). L'apprentissage et les tests ont été effectués uniquement sur les journées arrosées par la quantité correspondante.

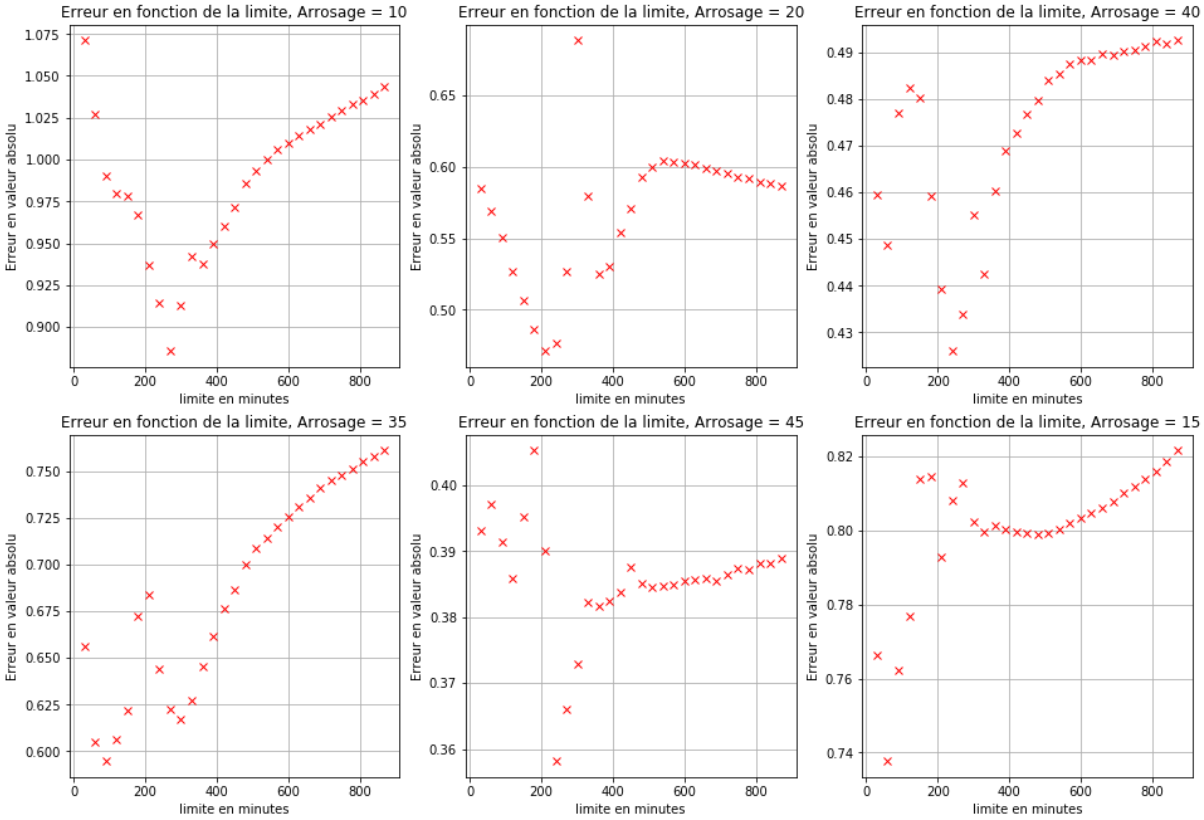

Figure 14 : Erreur en fonction de l'arrosage et de la limite utilisée entre la partie d'évaporation et la partie de stabilisation. Les données ont été séparées suivant la quantité d'arrosage. Par exemple, le graphique en haut à gauche n'utilise que les données avec un arrosage de 10 secondes. On peut voir que la limite ne dépend pas de la quantié d'arrosage. Une plus grande quantité d'arrosage ne signifie pas une période de stabilisation plus tardive.

On peut observer que la quantité de précipitation n'a qu'un très faible impact sur cette limite qui se situe en moyenne vers240 minutes après l'arrosage. Pour un arrosage de 35 et 45, deux minimums locaux existent. Il s'agit environ de 90 minutes et 240 minutes après l'arrosage. Seul le minimum de 240 minutes est conservé, car il s'agit du même que pour les autres arrosages. Cette même limite optimale est obtenue si on mélange tous les arrosages utilisés. Ainsi, la séparation à 240 minutes entre les parties d'évaporation et stabilisation sera la même pour tous les arrosages. J'utiliserai cette dernière pour la création de mon algorithme de prédiction, car elle me permet d'obtenir une bien meilleure précision.

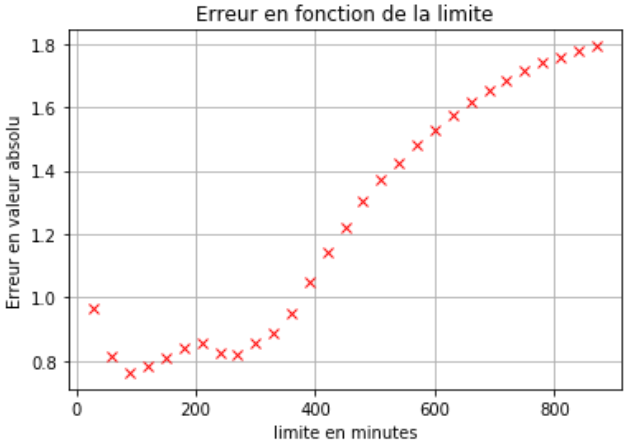

Figure 15 : Erreur en fonction de la limite utilisée pour toutes les valeurs d'arrosage rassemblées.

Analyse des résultats

Le but du projet est de fournir un logiciel permettant de définir un plan d'arrosage sur plusieurs jours pour atteindre un certain niveau d'humidité. La création du modèle de prédiction a été basé sur des données d'arrosage journalier. Un arrosage était programmé chaque jour. Dans le domaine du pratique, on ne veut pas toujours arroser chaque jour ou, si on désire fortement augmenter l'humidité en utilisant le moins d'eau disponible, on peut vouloir arroser plusieurs fois par jour. C'est pourquoi, mon logiciel, qui est présenté dans le chapitre suivant « *Elaboration d'un programme d'arrosage* », a la fonctionnalité d'arroser une ou deux fois par jour ou ne pas arroser du tout. Ce choix de se limiter à deux arrosages maximums par jour a été fait car il faut au moins 6 heures pour que l'humidité se stabilise correctement après un arrosage. De plus cette limitation permet de réduire l'utilisation d'eau. S'il faudrait arroser plus fréquemment pour atteindre une certaine quantité d'humidité, on peut considérer que la demande est disproportionnée ou que l'environnement de la culture n'est pas adapté à la culture de plantes.

Ainsi notre prédiction doit être correct après 12h, 24h et plus. Selon les résultats affichés précédemment, deux algorithmes obtiennent de très bons résultats. Il s'agit de la régression linéaire multiple avec la méthode des moindres carrés (LSMLR) séparé en trois parties et de la régression gradient boosting. Ce dernier obtient une très légère meilleure précision. Cependant, les tests ont toujours été effectués sur des journées comportant un unique arrosage. Aucune journée comportant aucun arrosage ou deux arrosages n'est disponible pour effectuer des tests.

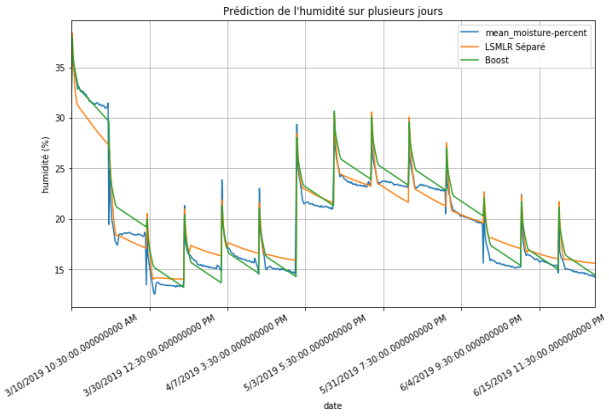

Figure 16 : Prédiction effectué à l'aide de LSMLR séparé en trois parties (en orange) et la régression Gradient Boosting (en vert). En bleu figure les valeurs réelles.

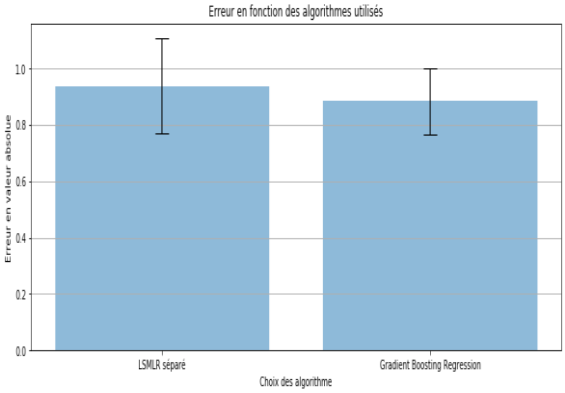

Figure 17 : Erreur de la prédiction par rapport à la valeur réelle avec LSMLR séparé en trois parties et la régression Gradient Boosting.

Si on observe l'évolution de l'humidité des données récoltées (voir graphique ci-dessous), la pente de l'humidité a tendance à être très faible en fin de journée. Ceci est également le cas avec la méthode LSMLR. A contrario, avec la régression gradient boosting, la pente est plus élevée en fin de journée qu'avec LSMLR. Ainsi, si on envisage qu'on arrosage qu'un jour sur deux, on peut imaginer et conclure que LSMLR aura une meilleure prédiction que gradient boosting. C'est pour cette raison que j'ai décidé d'utiliser pour mon logiciel la régression linéaire multiple. Nos futures prédictions auront donc tendance à être inférieur ou supérieur d'1% d'humidité.

Elaboration d'un programme d'arrosage

Le meilleur algorithme de prédiction et les meilleurs facteurs ont maintenant été sélectionnés. Les résultats obtenus permettent d'obtenir une précision très correcte. En moyenne l'erreur à une itération donnée est d'environ 1 % en valeur absolue. L'implémentation d'un logiciel fournissant la quantité d'eau qui est nécessaire pour arroser est enfin possible. Ce logiciel demandera à l'utilisateur de fournir le pourcentage d'humidité actuelle, le nombre de jours sur lesquels se déroule la prédiction, l'humidité désirée à la fin de cette durée et la température prévue pendant cette période. Ensuite le logiciel fournira la quantité d'arrosage journalière qu'il faut pour obtenir l'humidité désirée.

En ce qui concerne la température, j'ai fait le choix de demander à l'utilisateur uniquement la température pendant la journée et la température pendant la nuit. Une autre solution aurait été de prendre la météo fournie sur le web mais cette dernière correspond à la météo en extérieur et n'est pas du tout comparable à la température en intérieur. De plus, on ne demande qu'une simple estimation car le facteur température n'a pas vraiment d'impact. Une simple ouverture de fenêtre ou de store peut changer drastiquement la température.

Le logiciel fait le choix d'un arrosage homogène durant la période de simulation C'est-à-dire que l'arrosage aura de très faible variation entre chaque jour. L'arrosage de base est prévu pour s'effectuer à midi durant le fuseau horaire d'été ou à 11h durant l'heure d'hiver.

- S'il s'avère qu'un arrosage quotidien est insuffisant pour atteindre le taux d'humidité souhaité, le logiciel changera d'un seul arrosage quotidien à deux arrosages quotidiens. Dans ce cas, un arrosage se déclenchera à midi et un autre à minuit.
- S'il s'avère qu'il ne faut pas arroser, le système nous informe également.

La quantité maximale d'un arrosage est de 50 secondes. Cette limite a été fixée pour éviter les risques de fuite d'eau. De plus, par expérience, trop d'arrosage n'est pas efficace et le risque de perte d'eau est trop grand. Le but premier de cette expérience étant d'économiser de l'eau. Si malgré tout, deux arrosages quotidiens seraient insuffisants pour atteindre l'objectif d'humidité, le logiciel l'indique et fournit le plan d'arrosage qui s'y rapprocherai le plus avec deux arrosages journaliers.

Voici ci-dessous une image de l'utilisation du logiciel. Elle est précaire mais rempli entièrement les fonctionnalités souhaitées. Ce logiciel permet de définir la quantité d'arrosage uniquement pour le bac Déméter étant donné que seules ses données ont été utilisées pour la construction du modèle.

Voici ci-dessous, un exemple d'utilisation du logiciel et du résultat qui permet d'obtenir.

```
chabert@chabert-ZenBook:~/Documents/SensorProject/BSc-sensor-machine-learning$ python3 Prediction.py
Import de la database
Preparation des donnees ...
Calcul de la regression ...

Vous désirez prédire la quantité d'eau que vous devrez arroser dans le but d'atteindre un certain niv
eau d'humidité ? Notre algorithme est fait pour ça !
Pour y arriver, nous avons besoin de quelques paramètres.

Sur combien de jour ? : 3
Humidité actuelle ? : 21
Humidité souhaitée ? : 24

Pour le jour numéro :  1
Temperature pendant la journée ? : 31
Temperature pendant la nuit ? : 27

Pour le jour numéro :  2
Temperature pendant la journée ? : 30
Temperature pendant la nuit ? : 26

Pour le jour numéro :  3
Temperature pendant la journée ? : 31
Temperature pendant la nuit ? : 28

Meilleur arrosage journalier (1 arrosage par jour):  [40, 45, 45]
L'arrosage doit être fait à midi.
Humditité finale prédite :  23.9544771734
```

Figure 18 : Capture d'écran de l'interface logiciel. Le programme est lancé en exécutant la commande Prediction.py. Sur cette capture, on peut voir que la prédiction a été faite sur 3 jours et que l'arrosage recommandé est de 40, 45 et 45 secondes.

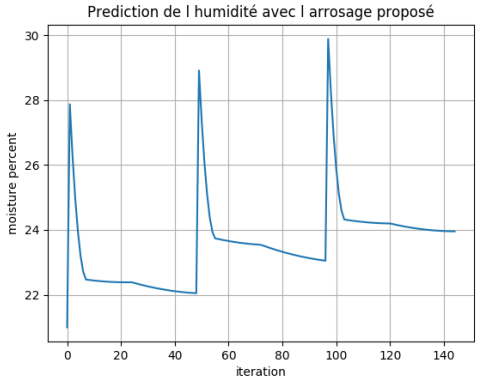

Figure 19 : Prédiction de l'humidité suite à l'arrosage recommandé.

Conclusion

Le logiciel créé remplit, avec une précision relativement correcte, les fonctionnalités escomptées. C'est-à-dire il nous renvoie un planning d'arrosage pour atteindre un certain niveau d'humidité du sol. Les résultats sont précis à +/- 1% d'humidité en moyenne par jour. On peut donc s'attendre à une plus grande distorsion pour un planning effectué sur une longue période. Par exemple, en moyenne, un planning sur 4 jours aura une différence de 4% d'humidité entre la prédiction effectuée et les résultats observés en partant que les paramètres de température insérés par l'utilisateur sont proches de la réalité. Etant donnée que la plage d'eau disponible dans le sol utilisé est d'environ 15%, ce pourcentage s'avère être dans une marge d'erreur relativement correcte. Le logiciel a tendance a surévalué un faible arrosage et à tendance à sous-évalué un arrosage conséquent.

Le programme fonctionne très bien quand certaines conditions sont remplies. A savoir que le logiciel fonctionne uniquement pour le bac Demeter puisque des choix sur les données ont dû être fait. Ce choix a dû être fait car des problèmes de récolte de données sont survenu dans le bac Cérès. Le programme peut être réutiliser pour d'autres bac sous réserve que les capteurs sont placés à une même distance d'un point d'arrosage et que les capteurs soient étalonnés entre eux.

Ce programme pourrait être améliorer de la façon suivante :

- En ayant plus de données d'apprentissage. La période de collecte de données aurait pu être plus longue. De plus il aurait fallu étalonner les capteurs d'humidité entre eux. En effet ceci aurait pu permettre d'utiliser les données de bacs différents. Un plus grand nombre de données d'apprentissage permet d'améliorer la précision.
- En ayant des données d'apprentissage sur une fréquence d'arrosage variables. En effet ceci aurait pu permettre de voir l'impacte de deux arrosages par jour sur l'humidité et répondre à la question « Est-ce que l'humidité réagit différemment à la fréquence d'arrosage ? »
- En approfondissant la question du bien-être de la plante. Une étude de l'impacte de l'humidité sur la croissance d'une espèce de plante aurait été intéressante. Pour ce faire, il y aurait fallu avoir plusieurs plants avec des arrosages programmables indépendants, à température et terre identique. Ceci aurait par exemple permis de répondre à la question « Est-ce que les épinards préfèrent une humidité de 15%, 20% ou 25% ? »

Bibliographie

Ahmad, Sajjad, Ajay Kalra, et Haroon Stephen. « Estimating soil moisture using remote sensing data: A machine learning approach ». *Advances in Water Resources* 33, n° 1 (1 janvier 2010): 69-80. https://doi.org/10.1016/j.advwatres.2009.10.008.

Goldstein, Anat, Lior Fink, Amit Meitin, Shiran Bohadana, Oscar Lutenberg, et Gilad Ravid. « Applying machine learning on sensor data for irrigation recommendations: revealing the agronomist's tacit knowledge ». *Precision Agriculture* 19 (2017): 421-44. https://www.semanticscholar.org/paper/Applying-machine-learning-on-sensor-data-for-the-Goldstein-Fink/46a0541b6c3ed4563a31ad395620531389548baa

Moorthy, Aadith. « Global Custom-Tailored Machine Learning of Soil Water Content for Locale Specific Irrigation Modeling with High Accuracy » 2, n° 10 (s. d.): 10, octobre 2016. https://ijoear.com/Paper-October-2016/IJOEAR-SEP-2016-34.pdf

Rakotomalala, Ricco. « Ridge – Lasso – Elasticnet », s. d., 41. Cours de l'université de Lyon2. https://eric.univ-lyon2.fr/~ricco/cours/slides/regularized_regression.pdf

Rebecca Shortt, Anne Verhallen, & Pam Fisher. (Janvier 2019). « Surveiller l'humidité du sol pour améliorer les décisions d'irrigation ». Site web du ministère de l'agriculture, de l'alimentation et des affaires rurales de l'Ontario. http://www.omafra.gov.on.ca/french/engineer/facts/11-038.htm.

Singh, Harshdeep. « Understanding Gradient Boosting Machines ». Towards Data Science, 3 novembre 2018. https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab.

Valo Martine. (2015). « La crise de l'eau illustrée en 5 graphiques ». https://www.lemonde.fr/ressources-naturelles/article/2015/03/20/la-crise-de-l-eau-illustree-en-5-graphiques_4597592_1652731.html.

« L'eau de la Terre : origine et répartition ». *Onglet Pédagogique 7e Continent* (blog). http://www.septiemecontinent.com/pedagogie/lesson/eau-terre-origine-repartition/.

« Planetoscope - Statistiques : Consommation d'eau dans le monde ». 2019. https://www.planetoscope.com/consommation-eau/239-consommation-d-eau-dans-le-monde.html.