الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة البليدة 1
Université de Blida 1

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Analyse d'un modèle de régression linéaire pour la prédiction des cas COVID-19

Mémoire de Master

Filière Électronique

Spécialité Instrumentation

présenté par

Boudouani Lilia

&

Gacem Rania

Proposé par : **Farid Ykhlef**

Année Universitaire 2019-2020

Remerciements

Ce présent travail de mémoire et le frais des efforts et des sacrifices concertés par moi et ma chère binôme, donc je tiens tout d'abord a la remercier énormément, car ce travail n'as pas pu aboutir que grâce a notre dieu tout puissant et l'effort de ma binôme GACEM RANIA.

J'adresse ensuite mes remerciement a notre encadreur YKHLEF FARID pour ses conseils, sa disponibilité, et sa compréhension.

Sans oublier les membres du jury pour avoir bien voulu donner de leur temps pour lire ce mémoire et faire partie des examinateurs.

Dédicaces

Je dédie ce modeste travail et ma profonde gratitude àma chère binôme «boudouani lilia» et A tous celui qui a sacrifié pour m'offrir les conditions propices à ma réussite :

A ma grande sœur « AICHA », à qui je dois la réussite, pour l'éducation qu'elle m'a prodiguée ; avec tous les moyens et au prix de tous les sacrifices qu'elle a consentis àmon égard, pour le sens du devoir qu'elle m'a enseigné depuis mon enfance.

A ma chère maman

A l'âme de mon père

Et toute ma famille avec tous mes sentiments de respect, d'amour, de gratitude ainsi je remercier vivement notre encadreur Mr YKHLEF FARID pour ses conseils et son suivi durant la réalisation de notre projet.

Aux êtres qui me sont chers, à mes parents, à ma petite sœur bien aimée à toute ma famille, et à tout qui m'ont souhaité le succès et le bonheur dans ma vie ….

Je dédie ce modeste travail.

Résumé

ملخص

تهدف هذه الدراسة إلى التنبؤ بسلسلة زمنية للمجاميع التراكمية المستقبلية لحالات COVID-19، باستخدام خصائص مثل الموقع الجغرافي وعدد البلدان/المناطق التي تم اكتشافها وتحليله، باستخدام لغة برمجة بايثون. ينشر نموذج التوقع أساليب تعلم الآلة الخاصة بجهاز الانحدار الخطي وآلة دعم المتجهات على قاعدة بيانات تم جمعها في جميع أنحاء العالم.توضح النتيجة اتجاهًا تصاعديًا لميزانية عمومية مدتها عشرة أيام، ولكنها تأخذ مسارين متميزين من خلال فحص توافق الحالات المتوقعة مع الحالات الفعلية اليوم.

كلمات مفاتيح :كوفيد-19 , طرق التعلم الآلي, تحليل السلاسل الزمنية ؛ وباء ؛ فيروس كورونا الجديد ,جهاز الانحدار الخطي.

Résumé

Cette étude a comme objectif de prédire une série chronologique des futurs totaux cumulatifs des cas de COVID-19, à l'aide des caractéristiques tels l'emplacement géographique et le nombre de pays/régions, mises à l'analyse et le traitement par le langage de programmation Python. Le modèle de prédiction déploie les techniques d'apprentissage automatique, de régression linéaire, et de machine à vecteurs de support sur une base de données récoltée dans le monde entier. Le résultat illustre une tendance ascendante pour un bilan de dix jours mais prend deux chemins distincts en examinant la conformité des cas prédits aux cas réels d'aujourd'hui.

Mots-clés : Pandémie ; nouveau coronavirus ; SVR ; COVID-19 ; méthodes d'apprentissage automatique ; analyse de séries chronologiques.

Abstract

The objective of this study is to predict a time series of future cumulative totals of COVID-19 cases, using characteristics such as geographic location and number of countries/regions, analyzing and processing by the Python programming language. The prediction model deploys machine learning, linear regression, and support vector machine techniques on a database collected from around the world. The result illustrates an upward trend for a ten-day review but takes two distinct paths when examining the conformity of predicted cases to actual cases.

Keywords: Pandemic; new coronavirus; SVR; COVID-19; machine learning methods; time series analysis.

Acronymes et abréviations

IA	:	Intelligence artificielle
SVR	:	Machine à vecteur de support pour régression
SVM	:	Machine vecteur à de support
KNN	:	k nearest neighbbors
RBF	:	Fonction de base radiale gaussienne
OMS	:	Organisation mondiale de la Santé
MSE	:	mean squared error
MAE	:	mean absolute error

Table des matières

Liste des figures

Liste des tableaux

Introduction générale

L'homme a toujours réussi à résoudre divers problèmes dans le monde réel, en utilisant des outils ou des concepts constamment soumis à un développement progressif. Pour cela depuis plusieurs années la technologie fait l'objet d'un accompagnement qui définit ces outils représentant toutes sortes de connaissances et de compétences utilisées pour la conception, dérivant ainsi des solutions cohérentes développées dans différents domaines.

Aujourd'hui, la technologie de l'intelligence artificielle a ouvert les portes d'un nouveau monde lucratif, grâce à des algorithmes pour modéliser le cerveau humain, afin d'avoir une résonance plus ou moins similaire à celui-ci. Disciplinée par l'apprentissage automatique où la machine apprend par elle-même en insérant plusieurs exemples pour comprendre le contenu d'une image par exemple, cette approche a facilité de nombreuses tâches dont la conduite de voitures (véhicules autonomes), la régulation automatique du trafic, la reconnaissance d'image pour le diagnostic des maladies, les Smartphones plus intelligents dopés avec cette technologie (reconnaissance faciale, prévisions météo) etc. On a compris s'il y a du numérique, il y a de l'IA.

L'intelligence artificielle a un potentiel énorme car elle contribue et contribuera de plus en plus dans presque tous les domaines, y compris médical, sur lesquels le diagnostic de maladies est cité, par exemple en croisant des milliards de données ou en analysant des images, des tests génétiques permettant de prédire le risque et le taux de survie pour, par exemple, certains cancers, applications de santé et objets connectés qui permettent aux patients de s'impliquer davantage dans leurs propres soins et au soignant d'assurer un suivi plus régulier et approfondi [2]. En se concentrant plus spécifiquement sur la crise actuelle de la COVID-19, l'IA joue un rôle dans plusieurs aspects de cette pandémie car depuis le début lorsqu'il s'agissait de détecter la COVID-19, une entreprise canadienne de Toronto (Blue Dot) a repéré, en utilisant l'analyse de texte techniques dans des réseaux comme Facebook, les 27 premiers cas suspects de pneumonie autour d'un marché aux animaux à Wuhan (Chine) le 31 décembre 2019, et alertent les autorités. Blue Dot découvre ce qui deviendra une grande pandémie mondiale 9 jours avant l'OMS, puis vint le diagnostic de la maladie un scan des poumons (radiographie) qui représente en fait un système de reconnaissance

d'image basé sur des réseaux neuronaux, puis repérer une personne fiévreuse dans une foule par analyse d'image à l'aide de caméras infrarouges.

On parle aussi de la propagation rapide du coronavirus qui a mis toute la population humaine en danger. Les cas qui étaient et sont jusqu'à présent en forte augmentation, avec plusieurs décès signalés chaque jour, étaient plus que suffisants pour étudier cette propagation afin de prédire les futures occurrences de transmission. Cette étude fait l'objet de notre mémoire dans laquelle un modèle d'apprentissage automatique sera construit, basé sur un algorithme d'entrainement afin d'approximer le nombre de cas futurs (confirmés, décès et le nombre de guérisons), visualiser l'impact de la COVID-19 sur la population mondiale pour mieux anticiper et contrer la pandémie de coronavirus.

Dans la première partie de ce mémoire, nous définirons d'abord l'intelligence artificielle ainsi que la COVID-19 et son impact sur la planète, afin de donner un aperçu de l'apport et de l'importance croissante de cette première sur cette maladie infectieuse. Dans la deuxième partie, nous nous concentrerons également sur l'apprentissage automatique en introduisant également les concepts de base qui le définissent. Et enfin une troisième partie consacrée à la conception et à la mise en œuvre du modèle.

Chapitre 1 COVID-19 et l'intelligence artificielle

1.1 Introduction

L'apparition soudaine du coronavirus a plongé le monde dans une crise majeure, d'autant plus qu'il est rapidement devenu incontrôlable et inévitable par les techniques de traitement précédemment utilisées. Ainsi, prévoir l'ampleur de l'épidémie de COVID-19 à l'échelle régionale et mondiale était important pour prendre les mesures nécessaires concernant les plans de préparation et les interventions d'atténuation [4] tels que : installations médicales, confinement, distanciation sociale, fermetures d'écoles, etc.

Grâce à l'analyse et à la visualisation de la propagation du virus dans le pays avec les cas confirmés, il a été possible de prédire les scénarios futurs de propagation de la COVID-19 en Chine, en Europe, au Moyen-Orient et dans le monde entier. Toutes les prévisions ont été faites de manière à connaître dans les prochains jours le taux de transmission de cette maladie, en tenant compte des données réelles de la série chronologique du virus [4].

Dans ce chapitre, nous développons de manière explicite ce qu'est l'intelligence artificielle, la pneumonie COVID-19 et l'historique de son apparition, en détaillant l'aide que l'AI a apportée jusqu'à présent contre ce virus et les projets en cours au moment de la rédaction de notre mémoire (vaccins et autres). Ceci est discuté en trois sections.

1.2 Introduction à l'intelligence artificielle

1.2.1 L'intelligence artificielle dans le passé

Les premières machines intelligentes apparaissent au tout début en 1921 dans des pièces théâtrales de science-fiction, puis c'est autour de chercheurs scientifiques d'imaginer des machines pensantes, et c'est sur ce sujet qu'Alan Turing, un mathématicien britannique, publie en 1950 un article intitulé ***Computing Machinery and Intelligence*** on qu'il propose un test pour savoir si une machine s'approche d'une intelligence similaire à l'intelligence humaine. Il s'agit d'un test empirique qui implique un interrogateur humain dans une pièce, un autre humain dans une deuxième pièce et une entité artificielle dans une troisième pièce. L'interrogateur est autorisé à communiquer à la fois avec l'autre humain et l'entité artificielle et avec un dispositif basé sur du texte tel qu'un terminal. Et sur la base des réponses aux questions posées par l'interrogateur, nous pouvons distinguer l'autre humain de l'entité

artificielle. Si l'interrogateur ne peut pas le faire, le test de Turing est passé et nous disons que l'entité artificielle est intelligente [5]. Ainsi, c'est à partir de 1956 que le terme intelligence artificielle fait l'objet de discussions, il est utilisé pour la première fois au Dartmouth College l'une des célèbres universités des Etats-Unis, puis les premières applications de l'IA apparaissent dès les débuts de l'informatique. Depuis, les programmes se sont vraiment améliorés car en 1957 le psychologue Frank Rosenblatt a inventé le premier réseau d'apprentissage utilisant un simple réseau de neurones (discuté dans le deuxième chapitre).

En 1980, c'est la naissance des algorithmes d'apprentissage et des systèmes experts où la machine tente d'effectuer un diagnostic comme un expert humain dans différents domaines, jusqu'en 1997, lorsque le premier logiciel de reconnaissance vocale a été publié et installé sous Windows.

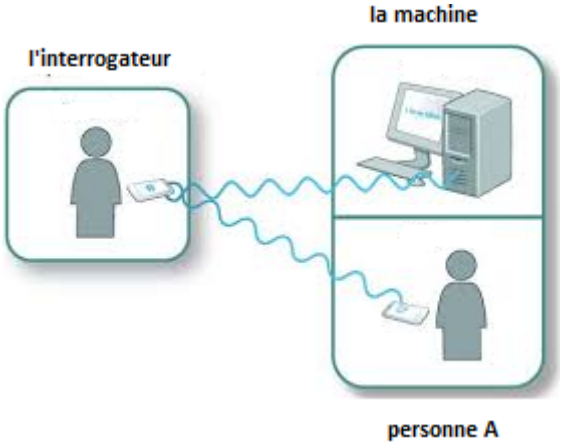


Figure 1.1Test de Turing.

1.2.2 Définition de l'IA

L'intelligence artificielle est un domaine de la science qui cherche à résoudre des problèmes logiques ou algorithmiques pour effectuer des tâches qui sont généralement effectuées par un humain. Les systèmes décrits ici sont principalement des systèmes informatiques qui traitent des données brutes, et à l'aide d'une série d'instructions (algorithmes) appliquées à ces données, un modèle est construit pour satisfaire un besoin spécifique.

Comme introduit précédemment, l'intelligence artificielle n'est pas nouvelle mais ces dernières années cette technologie a pris un grand tournant grâce à la croissance exponentielle des ordinateurs d'apprentissage et à la qualité des processeurs graphiques. Elle dispose désormais de données immenses (Big Data) et peut résoudre des problèmes très complexes, car plus nous fournissons de données aux machines, mieux elles

distingueront les caractéristiques qui composent cette tâche et mieux elle se comportera, c'est ainsi que fonctionne l'IA. L'IA imite l''humain dans ces capacités cognitives telles que la reconnaissance du langage, la reconnaissance de formes mais en plus de la connaissance humaine cognitive, l'humain a des fonctions supérieures telles que la pensée et l'éthique et c'est ici que deux types d'IA se différencient.

a) IA faible

L'intelligence artificielle faible est limitée car elle peut imiter une fonction spécifique parmi d'autres fonctions cognitives de l'intelligence humaine telles qu'un système de reconnaissance vocale, la reconnaissance faciale, etc. l'IA faible détermine la meilleure solution ordonnée par une règle mais si la réponse n'apparaît pas dans la liste des réponses déjà attribuées, elle ne peut pas résoudre le problème, on dit qu'elle n'est donc pas consciente d'elle-même.

b) IA forte

Contrairement à une IA faible, une intelligence artificielle forte est capable d'apprendre et de fabriquer sa propre solution pour imiter le fonctionnement du cerveau humain dans son ensemble, mais avec une conscience de soi en d'autres termes : comprendre ses actions. Plusieurs recherches sont actuellement en cours sur ce sujet pour construire des modèles avec une IA forte.

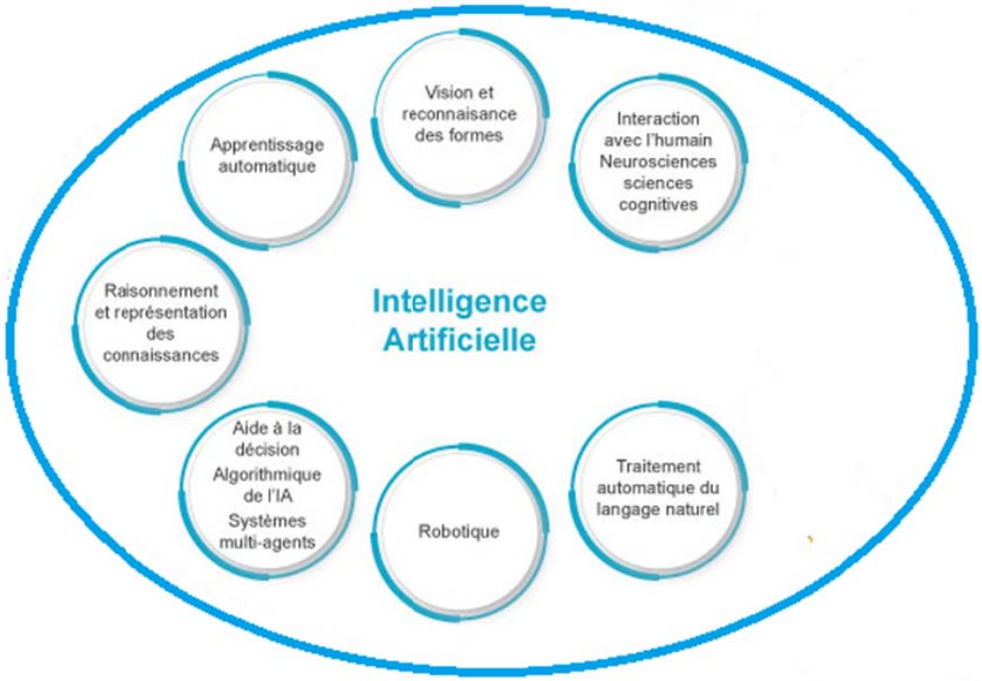


Figure 1.2Branches de l'intelligence artificielle.

L'intelligence artificielle est un domaine très vaste, une discipline mère qui à son tour recouvre plusieurs sous-disciplines ou sous-ensembles qui la composent, on cite : traitement du langage naturel pour l'intelligence liée à l'écriture et à la voix, robotique (manipulation de robots), raisonnement et représentation de connaissances car chaque système est régi par une règle pour attribuer une décision, une compréhension, une connaissance des formes, du machine learning, etc. Le schéma ci-dessous résume les branches les plus connues de l'IA.

L'apprentissage automatique est l'une des principales branches de l'IA car pour prédire ou effectuer une tâche, la machine doit apprendre. Pour cela, l'apprentissage automatique (ou Machine Learning en anglais) est identifié lorsque la machine apprend via ces soi-disant données d'apprentissage par des règles basées sur des approches statistiques et mathématiques, améliorant les performances du système pour ensuite effectuer quelque chose qui n'a pas été commandé (systèmes autonomes). Ceci est bien détaillé dans le chapitre à suivre.

1.3 La pandémie de la COVID-19

L'humain était toujours affronté face à des épidémies mortelles qui ont occasionnées plusieurs dégâts tel la peste le choléra la grippe espagnole, il y'on a de toute sortes mais l'émergence de ces épidémies étaient essentiellement causées par la transmission d'un agent pathogène de l'animal à l'humain, on les attribue le nom de maladies infectieuses (zoonoses). La famille des coronavirus en fait partie aussi.

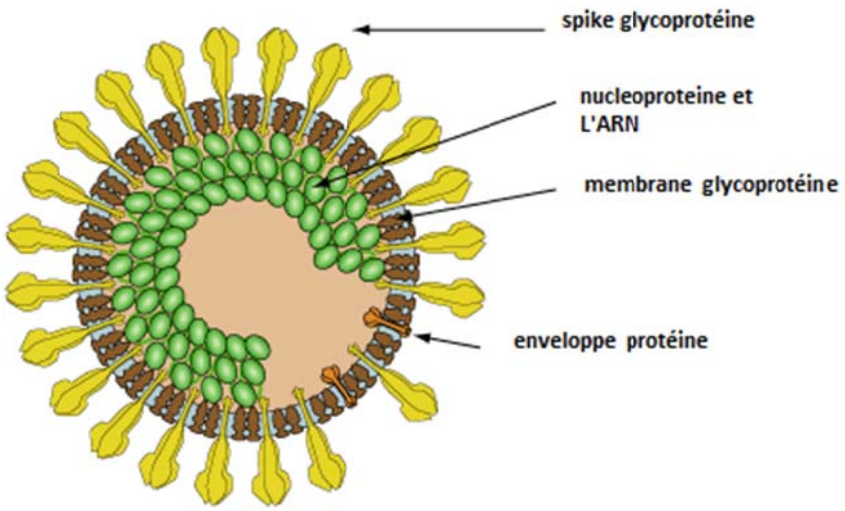


Figure 1.1Structure du coronavirus.

Le coronavirus vient du mot latin « corona » qui signifie couronne, ce nom est lié à l'apparence du virus, les différents types de ce dernier font partie des ARN simple brin à

polarité positive qui représente la nature du génome constituant le virus et plus exactement dans la famille des **Nidovirals** comme est 'il illustré dans la figure 1.3.

Remarque : L'ARN (acide ribonucléique) est un acide nucléique qui constitue la plupart des virus il est très proche chimiquement de l'ADN.

La famille des coronavirus est une vaste famille classée en quatre types :

- Alpha coronavirus
- Beta coronavirus
- Lignée A
- Lignée B
- Lignée C
- Gama coronavirus
- Delta coronavirus

1.4 L'intelligence artificielle une arme contre la COVID-19

Alors que la COVID-19 continu son accroissement dans le monde entier, l'intelligence artificielle a fait l'objet d'une arme très puissante qui a non seulement prédit l'émergence de cette pandémie, mais représenté aussi l'élément clé la détection rapide et précise des flambées épidémiques jusqu'aux prévisions politiques, sans oublier les robots conversationnels pour communiquer avec le public. L'intelligence artificielle et la science des données offrent de nouveaux moyens de lutter contre la pandémie [2].

1.4.1 Analyse de données assistée par l'IA -- Prédiction de la propagation de la COVID-19

En l'absence de médicaments ou de vaccins thérapeutiques pour le nouveau coronavirus, l'analyse des données et la prédiction sont tout aussi importantes, même plus importantes que le déploiement des techniques d'IA en affrontant le virus et en aidant au traitement. La COVID-19 est un ennemi qui nous ne pouvons connaître ses traits et comportements qu'en connaissance a posteriori à partir des données recueillies et des statistiques. Pour cela, nous élaborons deux approches que l'une introduira notre thème d'étude.

Depuis l'apparition de la COVID-19 et son augmentation alarmante, toute personne est devenue vulnérable à ses conséquences. Il était donc logique de développer un système qui puisse contrôler les ravages actuels et prédire intelligemment l'éclosion de la pandémie à l'échelle mondiale en simulant le nombre cumulatif des cas confirmés pour les prochains jours, à partir d'échantillons de données analysées sur 54 jours, ayant le but d'essayer de

freiner au maximum la pandémie ou se préparer au mieux aux dégâts qu'elle cause. Ce système s'appuie essentiellement sur l'emplacement géographique du pays et la visualisation des cas précédents dans une période précise, afin de pouvoir construire un modèle qui à partir des algorithmes d'analyse temporelle plus exactement les méthodes de régression, va prédire les valeurs des états susmentionnés pour les 10 prochains jours (Cette approche sera détaillée dans les prochains chapitres).

1.5 Conclusion

L'intelligence artificielle remplace ici le rôle du télescope en astronomie, un outil augmenté qui ne remplace pas l'humain, mais nous permet de traiter des choses impossibles à l'œil nu que ça soit pour la prévention ou pour le diagnostic précoce de la COVID-19. L'IA s'appuie essentiellement sur l'analyse de données et l'apprentissage automatique pour définir un modèle qui contribue à la lutte contre le virus d'aujourd'hui.

Chapitre 2 L’apprentissage automatique

2.1 Introduction

Depuis que l'intelligence artificielle a été reconnue comme une discipline au milieu des années 1950, l'apprentissage automatique a été un domaine de recherche central, car un système incapable d'apprendre peut difficilement être considéré comme intelligent. En effet, l'apprentissage offre une méthodologie potentielle pour la construction de systèmes à haute performance [18].

Aujourd'hui, les progrès technologiques encourageants ont exploité l'intelligence artificielle et l'apprentissage automatique pour aider à lutter contre la crise sanitaire actuelle à un rythme alarmant auquel elle se propage. Ils jouent un rôle important dans une meilleure compréhension et une meilleure gestion du nouveau coronavirus. La technologie d'apprentissage automatique permet aux machines (ordinateurs) d'imiter les capacités cognitives d'un être humain et d'ingérer de grands volumes de données collectées dans le monde entier pour identifier rapidement des modèles et des informations pour le contrôle et le traitement de la COVID-19

Dans ce chapitre, nous exposerons explicitement les branches du Machine Learning, les méthodes qu'il envisage pour construire un modèle et quelques critères de performance. Celui-ci est divisé en sections :

Section 2.2 : présentation générale de l’apprentissage automatique.

Section 2.3 : l’apprentissage supervisé.

Section 2.5 : Méthodes de classification de l’apprentissage supervisé.

Section 2.5 : Optimisation et Régularisation.

Section 2.6 : Conclusion.

2.2 Présentation générale de l'apprentissage automatique

2.2.1 Définition

L'apprentissage automatique est un ensemble d'algorithme consiste principalement à programmer des ordinateurs pour optimiser une performance critère utilisant des données d'exemple ou des expériences passées, permettant ainsi de construire un modèle général pour produire en sortie des réponses prédictives à de nouvelles données.

L'apprentissage automatique est l'intersection des statistiques, de l'intelligence artificielle et de l'informatique. Car, il repose d'une part sur les mathématiques, et en particulier les statistiques, en ce qui concerne la construction des modèles et leur inférence à partir de données, et d'autre part sur l'informatique, en ce qui concerne la représentation des données et la mise en œuvre efficace d'algorithmes d'optimisation [13].

2.2.2 Types d'apprentissage automatique

Selon l'objective recherché, ces dernières sont généralement classées en grandes catégories : l'apprentissage supervisé, l'apprentissage semi supervisé, l'apprentissage non supervisé et enfin l'apprentissage par renforcement.

a) Supervisé

L'apprentissage supervisé (ou Supervised Learning) est une forme d'apprentissage machine qui permette de construire des modèles à partir de données dont on connait le comportement ou la réponse, autrement dit : attributs étiquetés. Ces étiquettes supervisent l'apprentissage de l'algorithme [13]. En d'autres termes, l'apprentissage supervisé permet d'entraîner un modèle qui puisse imiter ce processus d'étiquetage, et qui puisse prédire pour une entrée x quelconque la valeur de la cible ou la réponse qui aurait normalement été donnée par un humain, pour ensuite être utilisés dans différentes applications, telles que la prédiction.

b) Semi supervisé

L'apprentissage semi-supervisé est une classe d'apprentissage automatique qui est constituée d'un jeu de données partiellement étiqueté, généralement d'une petite quantité de données dont on connait la réponse avec une grande quantité de données non marquées. L'avantage principal de cette approche c'est qu'elle permet d'éviter d'avoir à étiqueter l'intégralité d'exemples d'apprentissage, ce qui est pertinent quand il est facile d'accumuler des données mais que leur étiquetage requiert une certaine quantité de travail humain [13].

c) Non supervisé

Dans ce type d'apprentissage les données ne sont pas étiquetées, on fournit alors à la machine des exemples sans aucune valeur cible associée .En analysant la structure de ces données, la machine apprend elle-même à réaliser certaines tâches dont le clustering .Cela est réalisé , on regroupant les données d'entrées uniquement selon leur ressemblances en s'appuyant sur l'algorithme k-mean clustering ,cette technique est appelée aussi le partitionnement et elle est utilisée pour classer des documents , des photos ,etc.

Une autre tâche que peut réaliser la machine grâce à ce type d'apprentissage est la détection d'anomalie. Cette dernière consiste à trouver des échantillons dont les caractéristiques x sont éloignés de celles des autres échantillons. Cette technique peut être appliquée pour développer un système de surveillance par exemple.

d) Apprentissage par renforcement

Dans cette dernière catégorie les données d'entrée sont étiquètes. Par contre, l'apprentissage est guidé par l'environnement de manière à attribuer pour chaque action une récompense au système d'apprentissage, qui peut être positive si l'action était un bon choix, ou négative dans le cas contraire. Ainsi il faudra apprendre un modèle capable de prédire la meilleure décision à prendre étant donné un état de l'environnement.

2.3 L'apprentissage supervisé

2.3.1 Variables et data set

Les base de données (ou dataset en anglais) sont le matériau avec lequel les connaisseurs du domaine travaillent avec. Ce sont des enregistrements de mesure, ou des exemples de données d'une population d'individus ou d'objets Ω sujettes au problème d'apprentissage. On notera par ω un individu de Ω où Chaque individu de cette population sont associées deux catégories de variables ou d'attributs :

a) Attributs Exogènes

Il s'agit de l'ensemble des variables descriptives des individus. Dans le contexte de l'apprentissage supervisé, on les appellera également variables explicatives et elles seront notées $x_i, \dots, x_p$. Ainsi, l'ensemble des variables $x_i = (x_1, \dots, x_p)$peut être vu comme une application qui à tout individu ω ∈ Ω associe une valeur $x_i(\omega) = (x_1(\omega), \dots, x_p(\omega))$ prise dans un espace à p dimensions $D = d_1 \times \dots \times d_p$ [14].

b) Attributs Endogènes [14]

Il s'agit des variables à prédire. Contrairement aux variables exogènes, cette catégorie de variables, n'est identifiée que dans le contexte de l'apprentissage supervisé. On notera $y_i = (y_1, \dots, y_n)$variables endogènes.

Notant que ces variables peuvent être quantitatifs aussi bien que qualitatifs :

- **Variables quantitatifs (ou numérique) :** sont toutes les attributs qui se traduisent par un nombre entier ou réel.
- **Variables qualitatifs (ou catégorique) :** sont toutes les attributs qui exprime un état en modalités (catégories).

2.3.5 Formulation de l'apprentissage supervisé

Comme évoqué précédemment, l'apprentissage supervisé correspond au cas où l'objectif de construire un modèle est de prédire un certain résultat d'une entrée donnée sous la caractéristique d'être étiquetée .de manière analogue, le problème d'apprentissage supervisé est défini de la manière suivante :

Soit n observations de $x_i$sachant que$x_i = x_1, \dots, x_p$.Chaque observation $\{X_1, X_2, \dots, X_n\}$est un élément de l'espace des observations .

Ainsi on cherche à trouver une fonction $f: X \to Y$ telle que $f(x_i) \approx y_i$[13] .Où Y est l'espace contenanty_i.

2.3.2 Classification et régression

La nature de la cible à prédire dépendra principalement du type de problème à résoudre. Il s'agit d'une régression si cette dernière est de type quantitatif (continue), où un problème de classification si elle est qualitative (discrète). On définit ces deux types de problèmes d'apprentissage supervisé comme suit :

a) Régression

Dans un problème de régression, on cherche à mapper les données d'entrée à un attribut cible de valeur réel. En d'autres termes, il s'agit de prédire un résultat qui est continu, comme prédire l'âge ou le poids d'une personne par exemple.

b) Classification

Dans le cas d'une classification, l'attribut goal prend une valeur catégorique (sous forme de classe).C'est un choix qui se fait à partir d'une liste de possibilités prédéfinies (une

séquence de modalités). On a comme exemple la prédiction du sexe d'une personne (Homme/Femme), ou une détection de spam.

Quand le choix se fait entre deux différentes classes, on parle d'une classification binaire, autrement dit l'espace des étiquettes $Y = \{0, 1\}$.

Quand le choix se fait entre plusieurs possibilités, on parle d'une classification multi class, où l'espace des étiquettes $Y = \{1, 2, \ldots, C\}$ est fini et C est le nombre de classes [13].

2.3.3 Modèles paramétriques et non paramétriques

Pour construire un modèle idéale et développé toute en minimisant le taux d'erreurs, on doit principalement connaitre la manière dont le modèle va t 'il apprendre (en supervisant les données ou non), et le motif d'apprentissage (le type du modèle). Pour cela on distingue deux types de configuration :

a) Modèle paramétrique

Un modèle paramétrique est un modèle d'apprentissage qui est définie analytiquement sous la forme d'une fonction de décision, et dépend essentiellement de la Distribution de x_i et y_i (normal, gaussienne ou conditionnelle). Cette fonction est composée de paramètres représentant des éléments générés par l'ensemble de données, qui sont appris et ajustés pendant la phase d'entrainement de ce modèle avec des procédures ayant le but d'optimiser ces paramètres, elles seront abordées dans les sections à suivre.

Ces paramètres sont de taille fixe et diffèrent selon chaque modèle, les propriétés de l'ensemble de données et la tâche à accomplir, notant aussi que ces modèles apprennent très rapidement et nécessitent moins de données, citant par exemple un modèle de régression linéaire : $f(x) = \alpha x + \beta$, où f est la fonction de décision, α est le paramètre représentant l'intercepter de y et β le paramètre représentant la pente.

b) Modèle non paramétrique

Quand on est face à ce type de modèle la distribution de x_i et y_i est souvent inconnu, car ce dernier fait quasiment pas d'hypothèses sur les données. On ne peut donc pas définir une fonction de décision classique, notre modèle est ainsi déterminé à partir de l'ensemble de données, dont il en nécessite beaucoup pour apprendre. Ondit que l'apprentissage non paramétrique suit un processus itératif.

La flexibilité est l'un des plus grands avantages de ces modèles car ils peuvent s'adapter à un grand nombre de caractéristiques (features), sachant aussi qu'ils généralisent assez bien sur de nouvelles données. On a comme exemple l'algorithme KNN (k nearest

neighbors) basées sur l'ensemble de donnée d'entraînement, chaque nouvelle donnée prend l'étiquette des k les plus proches de cette dernière.

2.3.4 Espace d'hypothèses

Un espace d'hypothèses H est un ensemble de fonctions prédictives, décrites selon les caractéristiques (features traités) et le langage choisi qui n'est rien d'autre que le type du modèle représentant. Soit une régression linéaire paramétrique par exemple, ou une structure d'un neural network, etc. Une fois l'espace choisi, l'algorithme d'apprentissage va chercher la meilleure hypothèse h dans H de tel que l'erreur soit minime par rapport aux autres hypothèses. Pour cela l'évaluation de chaque hypothèse par une fonction de coût (ou loss function en anglais) est requise, car elle définit l'erreur de la fonction prédictive. On déduit ainsi que l'apprentissage peut être considéré comme un problème d'optimisation qui cherche à identifier une solution idéale vis- à- vis de la fonction de coût, dans un ensemble de solutions [15]. Ainsi la fonction prédictive choisie représentera la fonction de décision du modèle.

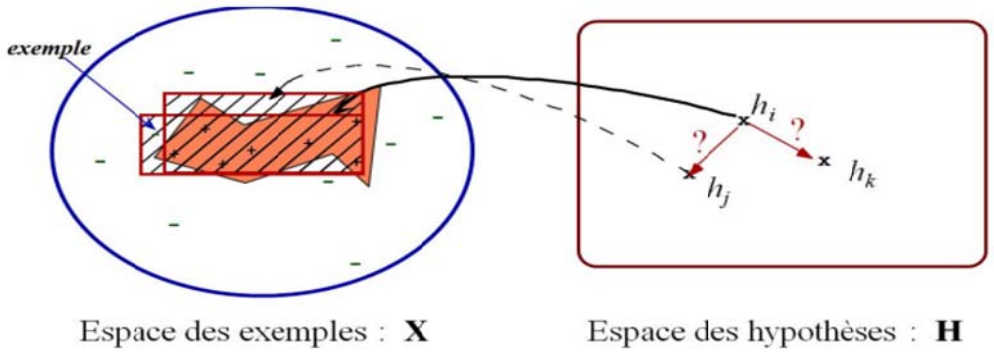


Figure 2.1 Évaluation des différentes hypothèses dans l'espace H.

2.4 Présentation des méthodes de l'apprentissage supervisé

2.4.1 Choix des méthodes de classification/régression

Le choix de l'algorithme va essentiellement être basé sur trois facteurs :

- Une bonne connaissance du problème à traiter et l'échéance de la tâche donnée.
- La visualisation des données qui consiste d'une part à connaitre la nature d'attributs qui se catégorisent en attributs qualitatifs ou quantitatifs et d'autre part la taille de données dans un cas d'étude.
- Les ressources et les connaissances disponibles acquises par les données (la nature de la distribution des attributs par exemple).

2.4.2 Méthodes de classification /régression

Les algorithmes de classification apprennent des données de façon à faire des prédictions proches d'étiquettes attribués de type qualitatives, et de généraliser le mieux possible sur de nouvelles base de données.

On distingue d'une part des méthodes discriminantes, qui consistent à estimer directement la classe d'appartenance d'un attribut citant :

- L'algorithme SVM (machine à vecteurs de support).
- La régression logistique.
- Réseaux de neurones.

Et d'autre part des méthodes génératives, qui consistent à estimer la probabilité d'appartenance de chaque classe. C'est des modèles statistiques qui s'appuient généralement sur les lois de probabilité citant :

- L'algorithme de naïve bayes

Il existe également d'autres algorithmes qui ne font référence à aucune de ces deux approches citant :

- L'algorithme KNN (k nearest neighbor)

Contrairement aux méthodes de classification, les algorithmes de régression simulent la relation entre les données d'observation et la variable cible de type quantitative, la fonction continue qui décrit l'algorithme diffère selon les exigences susmentionnées

Ces algorithmes varient des plus simples citant :

- La régression linéaire
- La régression polynomiale (non linéaire)

Aux plus compliqués :

- L'algorithme SVR (Les machines à vecteur de support pour la régression)
- Réseaux de neurones pour la régression (regression neural network en anglais)

Notant qu'il Ya plusieurs autres méthodes de régression/classification qui ne sont pas décrites. Mais pour ce mémoire nous nous limiterons à ce nombre d'algorithmes, la figure si dessous résume sous forme de carte tous les algorithmes du Machine Learning.

Figure 2.1 Les algorithmes du Machine Learning.

2.4.3 Régression linaire et polynomiale

a) Régression linéaire

La régression linéaire est la méthode de régression la plus simple où l'algorithme établie sous forme de droite la relation entre les observations et les variables étiquetées, en d'autres mots une somme pondérée de descripteurs [13] qui se rapproche le plus possible d'un ensemble de points. Elle est formulée de la manière ainsi :

$$z = \beta_0 + \sum_{i=1}^{n} \beta_i x_i$$

$$z = \sum_{i=0}^{n} \beta_i x_i$$

Pour idéaliser cette droite on doit minimiser la distance entre cette dernière par rapport au jeu de données, de manière à ce que cette ligne passe sur tous ses points il faudrait ainsi chercher des paramètres adéquats pour améliorer constamment notre modèle car on ne peut effectivement pas changer de dataset. Ces paramètres représente respectivement l'ordonné à l'origine (intercept en anglais) qui symbolise la hauteur de cette droiteβ_0 et la pente qui correspond au niveau de son inclination, puisque on a caser le jeu d'entrainement

en n observation chaqueβ_i est assigné à un vecteur $x_i$pour former une fonction de décision approprié au probléme.

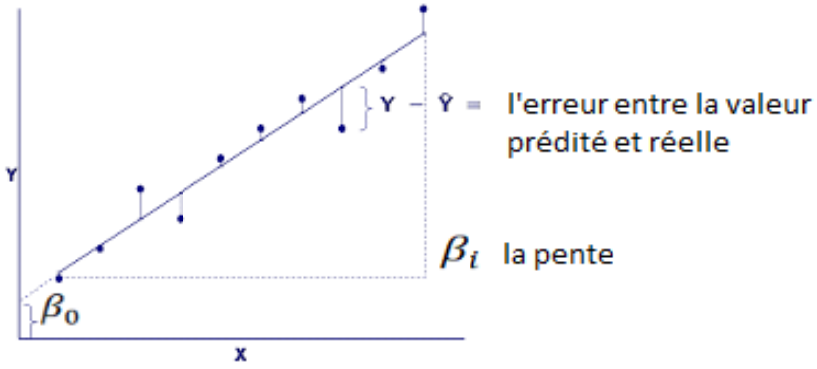


Figure 2.2 Droite de régression linéaire.

b) Régression polynomiale

Dans certaines tâches d'apprentissage, Il est parfois difficile d'ajuster le modèle en suivant la forme d'une droite de ce fait la régression polynomiale est une approche qui intègre des polynômes pour simuler sous la forme d'une courbe la relation entre les données d'observations et la réponse qui n'est généralement pas linéaire. Dans le cas de la régression polynomiale de degré d, on cherche une fonction de décision de la forme suivante [13] :

$$z = \beta_0 + \sum_{i=1}^{n} \beta_{1i} x_i + \sum_{i=1}^{n} \beta_{2i} x_i^2 + \cdots + \sum_{i=1}^{n} \beta_{di} x_i^d$$

Il s'agit en fait d'une régression linéaire sur $p \times d$ variables : $x_1, x_2, \ldots, x_n$, $x_1^2, x_2^2, \ldots, x_n^2$, $x_1^d, x_2^d, \ldots, x_n^d$ [13].

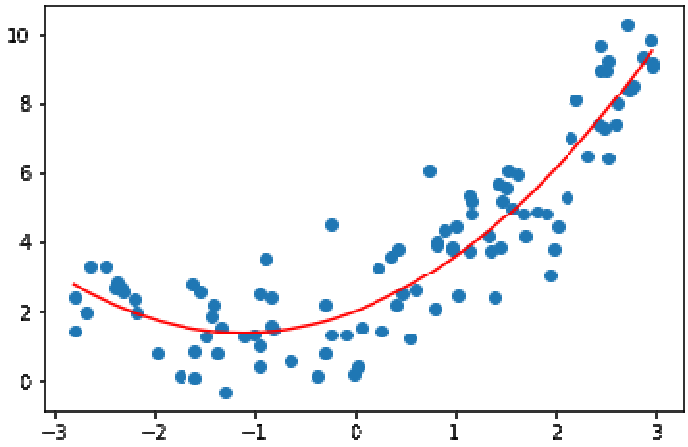


Figure 2.3 Régression polynomiale

2.4.4 Régression logistique

A raison de compléxité de l'algortihme nous nous limiterons pour notre étude au cadre de la régression logistique binaire, où la variable y_i prend deux modalités possibles. Commeintroduit précédemment La regression linéarie prédit grace à une droite ,la réponse d'une certaine tache donnée ou la cible peut prendre une infinité de valeurs .C'est un modéle ou nos predictions sont des variables qualitatives continus et c'est là que le probléme se pose, car dans un cadre de classification initialement bianaire ou y prends seulement deux valeurs $\{0,1\}$ construire un modèle linéaire n'est pas une bonne idée car on disposant par exemple d'attribut qui est relativement très positif ou très négatif engendre un resultat qui dépasserai fortment la limite du probleme la solution donc est d'ecraser l'ensemeble du jeu d'entrainement pour que l'eventail des possibilitées soit compris entre zéro et un d'un sens propre la fonction précédente requit une transformation à partir d'une loi logistique (fonction de sigmoide):

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad \text{ou} \begin{cases} z \to \infty, \sigma(z) \to 1 \\ z \to -\infty, \sigma(z) \to 0 \\ z = 0, \sigma(z) = 0.5 \end{cases}$$

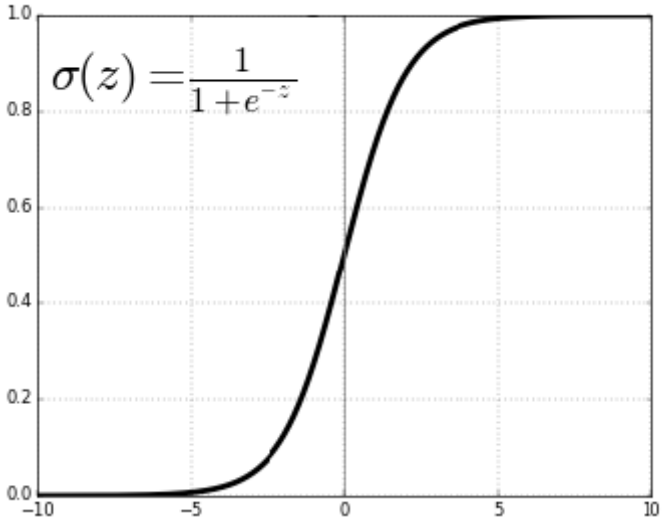


Figure 2.4 Fonction sigmoïde.

La fonction de décision de notre modèle n'est rien donc que :

$$f(x) = \widehat{Y} = \sigma(\sum_{i=0}^{n} \beta_i x_i)$$

Basé sur la fonction logistique , $\hat{Y}$ peut-être interprété comme la distribution conditionnelle de y_i sachant x_i:

$$p(y_i|x_i) = f(x)^{y_i}\ (1 - f(x))^{1-y_i}$$

Car lorsque y_i est égale à zéro : $p(y_i|x_i) = 1 - f(x)$ et lorsque c'est le cas contraire $p(y_i|x_i) = f(x)$.

Apprendre cette fonction revient à optimiser la paramètres qui la caractérisent (βdans notre exemple) et afin de trouver l'optimal desβ_i un algorithme locale est appliqué il s'agit du gradient descente qui avec l'approche de la vraisemblance ces derniers sont ajustés .

Tout d'abords calculons la vraisemblance de β :

$$l(\beta) = p(y_i|x_i\,\beta)$$

Puisque on dispose de n observation on obtient :

$$l(\beta) = \prod_{i=1}^{n} p(y_i|x_i\beta)$$

$$l(\beta) = \prod_{i=1}^{n} f(x)^{y_i}(1 - f(x)^{1-y_i}$$

Ainsi trouver le paramètre adéquat revient à maximiser la vraisemblance de ce dernier et puisque les probabilités sont toutes positives car $y_{i\in}[0,1]$ on se cerne à maximiser le log de cette fonction c'est-à-dire :

$$\log l(\beta) = \sum_{i=1}^{n} [y_i \log(f(x) + (1 - y_i)\log\,(1 - f(x))]$$

Maximiser cette fonction concave via l'algorithme du gradient descente est effectué en dérivent le terme précédent par rapport à la valeur initialement attribué à β pour l'actualisation du paramètre :

$$\beta(opt) = \beta(int) + \alpha\ \nabla_\beta l(\beta)$$

$$\beta(opt) = \beta(int) + \alpha\ \frac{\partial l(\beta)}{\partial \beta}$$

(On noteα est le taux d'apprentissage).

Comme la descente du gradient appliqué dans les réseaux neuronales (section précédente) cette algorithme est itératif car les β_i seront ajustés progressivement et lorsque

le local maximum du graphe de notre fonction concave est atteint on dira que $f(x)$ est égale par défaut à l'étiquettey_i .

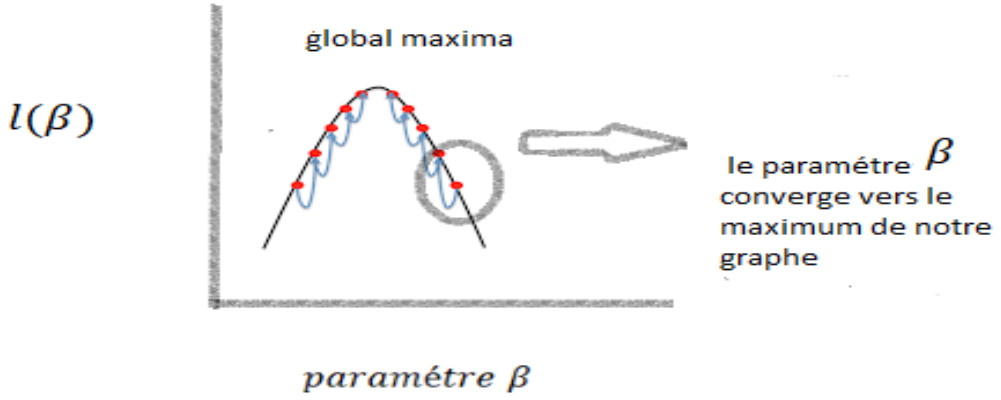


Figure 2.5 Monté du gradient.

2.4.5 Machine à vecteurs de support (SVM)

Les machines à vecteurs de support sont une famille d'algorithmes d'apprentissage définis pour la prévision d'une variable endogène qualitative initialement binaire [14], pour ensuite être généralisés aux problèmes à plus de deux classes. Les SVM sont basés essentiellement sur la recherche d'un hyperplan, ou ligne de meilleur ajustement entre deux classes, cette ligne représente la frontière de décision.

a) SVM linéaire

Dans le cas linéaire et on se place dans le cadre d'une classification binaire, de nombreuses droites peuvent séparer les classes, mais intuitivement certains d'entre eux sont meilleurs que d'autres. Ainsi, la droite optimale est celle qui maximise l'écart entre deux classes pour une capacité de généralisation idéal, et c'est ici qu'on introduit le terme de la marge : la marge γ d'une ligne séparatrice est la distance de cette droite à l'observation du jeu d'entraînement la plus proche [13]. Mais on est souvent face à un problème de plus de deux dimensions ($x_i \in \mathbb{R}^p$), où la droite est un hyperplan séparateur, et la marge γ est défini comme l'écart entre cet hyperplan séparateur et le point le plus proche appartenant à la classe A ou la classe B. De ce fait, on distingue deux autres hyperplans parallèles à ce dernier où chaque hyperplan contient au moins une observation de classe, ces observations sont nommées vecteurs de support, car elles soutiennent les hyperplans et n'importe quel changement au niveau de ces vecteurs impactera la position de l'hyperplan séparateur.

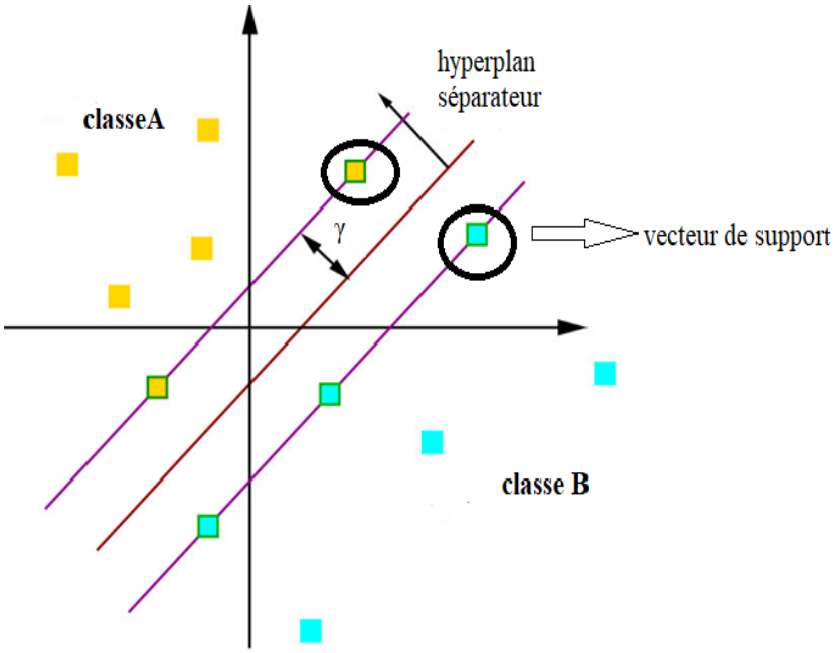


Figure 2.6 La résolution d'un problème de classification binaire aux données linéairement séparable avec l'algorithme SVM.

L'équation de l'hyperplan séparateur est un produit scalaire sous la forme $\langle \vec{w}, \vec{x} \rangle + b = 0$ [2] ou $w^T x + b = 0$ [16], ces deux équations représentent géométriquement parlent ,l'idée de projeter notre donnée d'entrainement sur le vecteur de poids w (un paramètre caractérisant le SVM linéaire et qui est orthogonal à l'hyperplan), de façon à connaitre si on est placé dans la classe A ou B par ce produit scalaire. Le signe de cette quantité va déterminer alors la résolution de notre problème dans un cadre de classification à deux classes. En d'autre termes, on définit une fonction $h(x) = signe\ (w^T x + b)$ de tel que (x_i, y_i) appartenant à la classe A, si $h(x)$ est du même signe que y_i alors x est classée du côté droit de l'hyperplan (classe A) si non elle appartiendra à l'autre côté de ce dernier, Notant que b est le déplacement par rapport à l'origine.

Comme énoncé précédemment le but essentiel ici est de maximiser la distance marginale entre ces deux hyperplans mais par optimisation sous contrainte, l'idée est que l'hyperplan doit être le plus loin possible des points de l'entrainement des deux côtés. Pour cela, on suppose que x_1 est le vecteur support appartenant à la classe A, et x_2 est le vecteur support appartenant à la classe B. On définit ainsi respectivement les deux équations des hyperplans parallèles :

$$\begin{Bmatrix} w^T.x_1 + b = +m \\ w^T.x_2 + b = -m \end{Bmatrix}$$

Sachant que la distance entre x_1 et x_2 est équitable on normalise la constante de la première catégorie comme étant positive et la deuxième comme étant négative, et comme nous sommes également libres de réévaluer T,$\|w\|$, et m, on peut choisir la constante m comme étant égale à 1:

$$\begin{Bmatrix} w^T.x1 + b = +1 \\ w^T.x2 + b = -1 \end{Bmatrix}$$

On soustrait ensuite les deux équations pour déterminer la marge. Géométriquement parlent, quand on projette cette distance sur le vecteur $w$qui est normalisé on aura un vecteur de longueur égale à 2γ:

$$\frac{w^T}{\|w\|}(x1 - x2) = \frac{1}{\|w\|}$$

Donc pour que l'hyperplan soit le plus optimal possible on cherche à maximiser cette valeur de tel que :

$$y_i \begin{pmatrix} 1 & w^T x + b \geq 1 \\ -1 & w^T + b \leq -1 \end{pmatrix}$$

Ainsi, de manière générale la condition d'optimisation se manifeste comme étant le $max \frac{1}{\|w\|_2}$ sous les n contraintes$y_i \times (w^T x + b) \geq 1$ [2].

Un SVM robuste est un classificateur qui est relative à des petites perturbations des donnés d'entrainement. Car effectivement, les observations ne sont pas toute a fait linéairement séparable en cas réelle. Notre but est de trouver un compromis entre les erreurs de classification et la taille de la marge [13], de cela on introduit un terme d'erreur à la valeur de cette dernière :

$$argmin \frac{1}{2}\|w\|_2 + C\sum_{i=1}^{n} \xi_i$$

Sous contraintes : $y_i \times (w^T x + b) \geq 1 - \xi_i$

Où Cest Le paramètre de régularisation qui est essentiellement la mesure dans laquelle nous voulons éviter les erreurs de classification (nombre d'erreurs), et $\boldsymbol{\xi}$ représente la valeur de l'erreur. On note aussi que si Cest très petit, l'optimiseur trouvera le classificateur de marge maximum même s'il a mal classé certains points.

Pour des raisons de simplification on s'y intéresser à une classification à deux. Un problème de classification multi-classe peut être résolu en traitant préalablement plusieurs

sous-problèmes de classification binaire, puis en combinant le résultat des différentes fonctions de décision.

b) SVM non linéaire

Dans le cas où nos données sont dans un espace où une séparation linéaire est impossible, aucun hyperplan ne peut être attribué. On fait donc l'introduction de la méthode des noyaux, qui consiste pratiquement à convertir nos données en un espace intermédiaire de dimension plus élevé permettant de réécrire notre jeu d'entrainement de manière à ce qu'il soit linéairement séparable par une transformation$\boldsymbol{\varphi}$ comme indiquée dans la figure 2.8, Puis de le reconvertir à son état initiale une fois le problème résolu.

Analytiquement parlent, si on redéfinit l'équation de l'hyperplan séparateur dans le cas linéaire (qui n'est pas approprié dans ce cas) mais sous la forme dual c'est-à-dire : $\sum_{i=1}^{n} \propto_i y_i x_i^T x + b = 0$ où$w^T = \sum_{i=1}^{n} \propto_i y_i x_i^T$ la transformation se fait de manière à aboutir à une équation qui peut être linéairement séparable mais dans un espace $\mathbb{R}^m$ où $m > p$:

$$\sum_{i=1}^{n} \propto_i y_i \varphi\left(x_i^T\right) \varphi(x) + b = 0$$

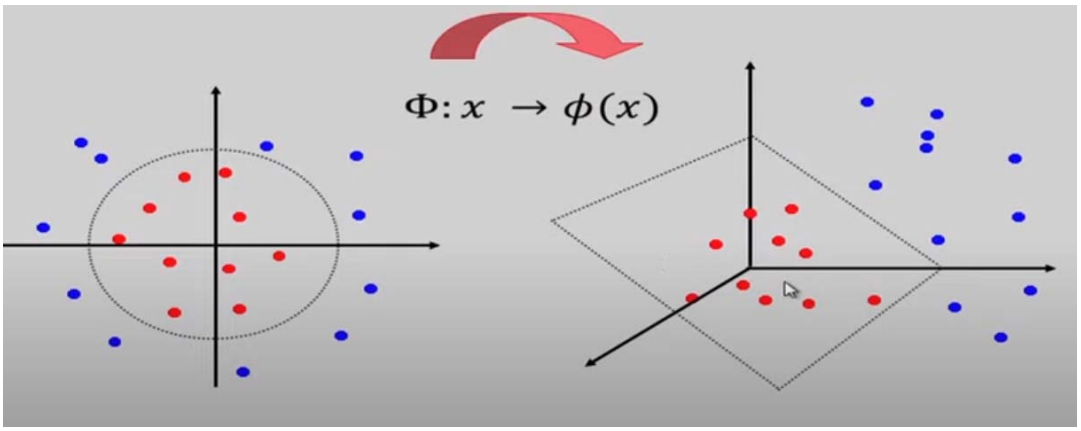


Figure 2.7 Le passage d'un espace $\mathbb{R}^p$en un espace $\mathbb{R}^m$par la transformation φ.

Mais trouver la forme explicite de φ est un problème très difficile à résoudre, car les calculs risque d'être très complexe (grande dimension).Pour cela, on fait appel à la fonction noyau (ou kernel function en anglais) qui est égale au produit$\varphi\left(x_i^T\right).\varphi(x)$ mais qui est très aisé à manipuler, il nous suffit de connaître le noyau k [13]:

$$k\left(x_i^T, x\right) = \varphi\left(x_i^T\right).\varphi(x)$$

La figure ci-dessous illustre une séparation de deux classes dans$m$ dimension une fois l'astuce de noyau était effectuée.

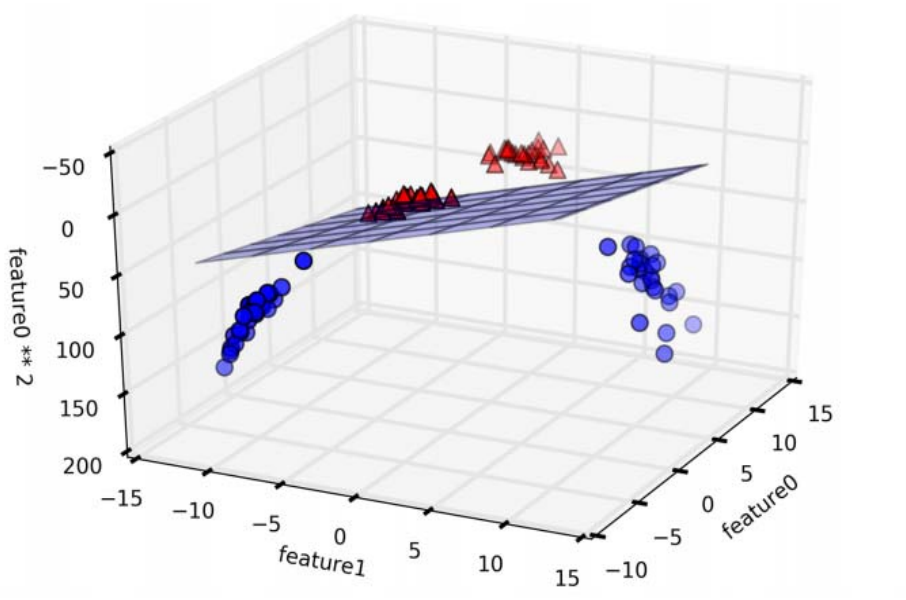


Figure 2.8 Frontière de décision d'un SVM non linéaire [12].

2.4.6 Machine à vecteur de support pour la régression (SVR)

Les SVR sont des modèles de régression basées essentiellement sur les éléments qui caractérisent les machines à vecteur de support, utilisés pour la prédiction d'une variable cible continu. Par conséquent, au lieux d'essayer d'adapter l'écart entre deux classes en maximisant la marge toute en essayant de réduire les cas qui se retrouvent dans la zone d'indécision (entre les deux hyperplans) ou même du mauvais côté, l'algorithme SVR va considérer au contraste que l'hyperplan séparateur adéquat est celui qui comptera le plus de points sur sa surface c'est-à-dire un maximum de points sur les deux hyperplans parallèles. Mais vue que la variable d'intérêt est réelle, il très difficile de prédire l'information car cette dernière pourra prendre une infinité de valeurs, pour cela une marge de tolérance est fixée (epsilon). L'idée principale est toujours la même : minimiser l'erreur, individualiser l'hyperplan qui maximise la marge, en gardant à l'esprit qu'une partie de l'erreur est tolérée [19].

Pour toute valeur en dehors d'epsilon ε , nous pouvons indiquer son écart par rapport à la marge comme ξ. L'objectif est de réduire ces écarts au maximum. L'algorithme SVR donne ainsi la flexibilité de définir la quantité d'erreur acceptable du, et trouver par conséquent une ligne appropriée pour adapter les données.

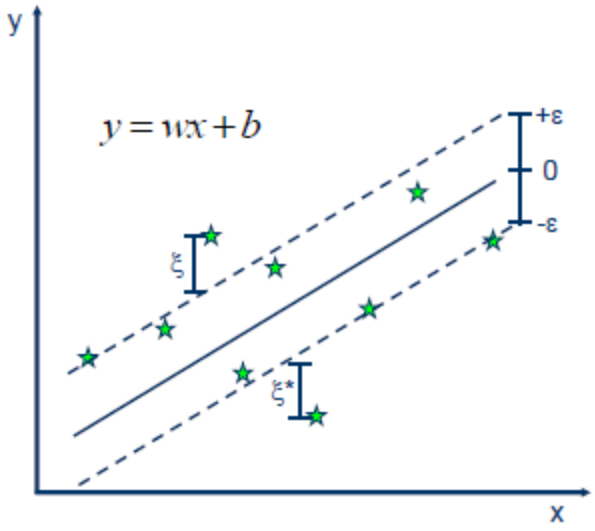


Figure 2.9 SVR linéaire [19].

Comme pour les SVM on cherche à minimiser la norme w et le terme ξ qui correspond cette fois-ci non pas aux erreurs de classification mais aux données qui divergent en quelque sorte et dépasse la marge de tolérance appelé zone insensible ε.

$$argmin\frac{1}{2}\|w\|_2 + c\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$

Avec les contraintes :

$$y_i - w^T - b \leq \varepsilon + \xi_i$$

$$w^T + b - y_i \leq \varepsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \geq 0;\ i = 1, \dots, i = n$$

2.5 Optimisation et régularisation

2.5.1 Optimisation

On rencontre principalement deux types d'optimisation :

a) Avec contraintes

En générale la solution à ce type d'optimisation est soumis à des restrictions sur les variables ces restrictions doivent être impérativement considérable et satisfaite.

Formulation de l'optimisation avec contrainte :

$$(max|min)\ f(x)$$

Sous contraintes :

$c_e(x) = 0$ Pour tout $e \in \{1, \dots, n\}$

$c_i(x) \leq 0$ Pour tout $i \in \{1, \dots, m\}$

$f(x)$ est la fonction d'objective qui représente une variance à maximiser ou une fonction de coût à minimiser. On note $c_i(x)$: vecteur représentant l'ensemble de conditions attribuées$(c_1(x), \dots, c_m(x))^T$.

b) Sans contraintes

Lorsque on est face à un problème d'optimisation sans aucune contrainte où on peut modéliser les n variables u représentant les paramètres du système dans toute l'espace sans limitation, plusieurs techniques sont mise en œuvre pour attribuer à ce dernier une solution plus au mois approché de l'idéale (zéro erreur est presque impossible) or la recherche des maxima ou minima d'une variance ou d'une fonction de coût est la technique la plus pertinente pour résoudre ce problème et ceci en employant un algorithme local d'optimisation. Citant, à titre d'exemple, l'algorithme du gradient qui est basé sur la dérivé de la fonction à optimiser par rapport au variable à ajuster, également l'algorithme du gradient stochastique.

2.5.2 Régularisation

Un modèle classé d'optimal est également un modèle qui fonctionnera activement avec des intrants en dehors de l'ensemble d'entrainement or ce n'est pas évident en cas de sur-apprentissage, où l'algorithme est considéré comme instable car un léger changement de son entrée changera sa sortie. L'instabilité de ce modèle est due au nombre de variables [13] assigné aux attributs exogènes, de ce fait le modèle devient très complexe. La régularisation vient donc utiliser un terme de pénalité ajouté à la fonction d'objectif ou de perte afin de contraindre certains paramètres du modèle pour réduire l'erreur de généralisation (erreur de test) [17].

Avant d'introduire le principe de régularisation, on définit quelques termes évoqué ci-dessus nécessaire à connaitre dans le Machine Learning dont on mentionne la généralisation et le phénomène du sur- apprentissage, sous-apprentissage :

- On appelle généralisation la capacité d'un modèle à faire des prédictions correctes sur de nouvelles données, qui n'ont pas été utilisées pour sa formation car notre but est de construire un modèle robuste avec de faibles erreurs mais qui puisse en outre prédire des

réponses justes lorsque on est face à un nouveau groupe d'individus en partitionnant les exemples disponibles par la méthode de validations croisées [13].

- La validation croisée est une méthode qui permet de diviser le jeu de données en proportions d'exemples auquel une partie est réservé à la phase d'apprentissage dont la machine va s'appuyer et une autre pour les observations restantes réservés à l'échantillon test. Pour effectuer ceci trois protocoles en étaient mis en œuvre ayant le but d'estimer l'erreur de généralisation on nomme le holdout cross validation qui le plus simples des protocoles.
- Le holdout consiste à séparer le jeu de données en deux parties : l'une sera utilisée pour construire le modèle ; l'autre pour le tester. Il faut juste définir la taille de chacun de ces deux échantillons. Généralement on utilise 70% des individus pour construire le modèle et 30% pour le tester. On parle alors de protocole "70/30". Le principal inconvénient de cette méthode est qu'il oblige à se passer de beaucoup d'individus pour la construction du modèle, et dans le même temps tous les individus ne sont pas exploités pour le test [14].
- k-fold corss validation va résoudre ce problème car c'est un protocole qui consiste à couper le jeu de départ en k subdivisions d'effectif équivalent par tirage aléatoire sans remise. Un modèle est ensuite construit sur k−1 subdivisions, et tester sur la subdivision Restante. Ce procédé est répété pour que chaque subdivision se retrouve une unique fois en test. Ainsi chaque individu s'est retrouvé une fois en test sur un modèle où il n'a pas été utilisé pour la construction [14].

a) La régression ridge

La Régression ridge est la fonction qui se formule de la manière suivante :

$$R(w) = \lambda \|w\|_2^2$$

Une fois ce terme rajouté à l'erreur estimé durant l'entrainement (fonction de perte) la nouvelle fonction d'optimisation revient maintenant à minimiser :

$$j_{ridge}(w) = \sum_{i=1}^{n} (y_i - \hat{Y}_i) + \lambda \|w\|_2^2$$

La norme $\|w\|_2^2$ est la distance euclidienne des p valeurs du vecteur w des n observations de x_i:

$$\|w\|_2 = \sqrt{w_1^2 + w_2^2 + \cdots + w_p^2}$$

$$\|w\|_2^2 = w_1^2 + w_2^2 + \cdots + w_p^2$$

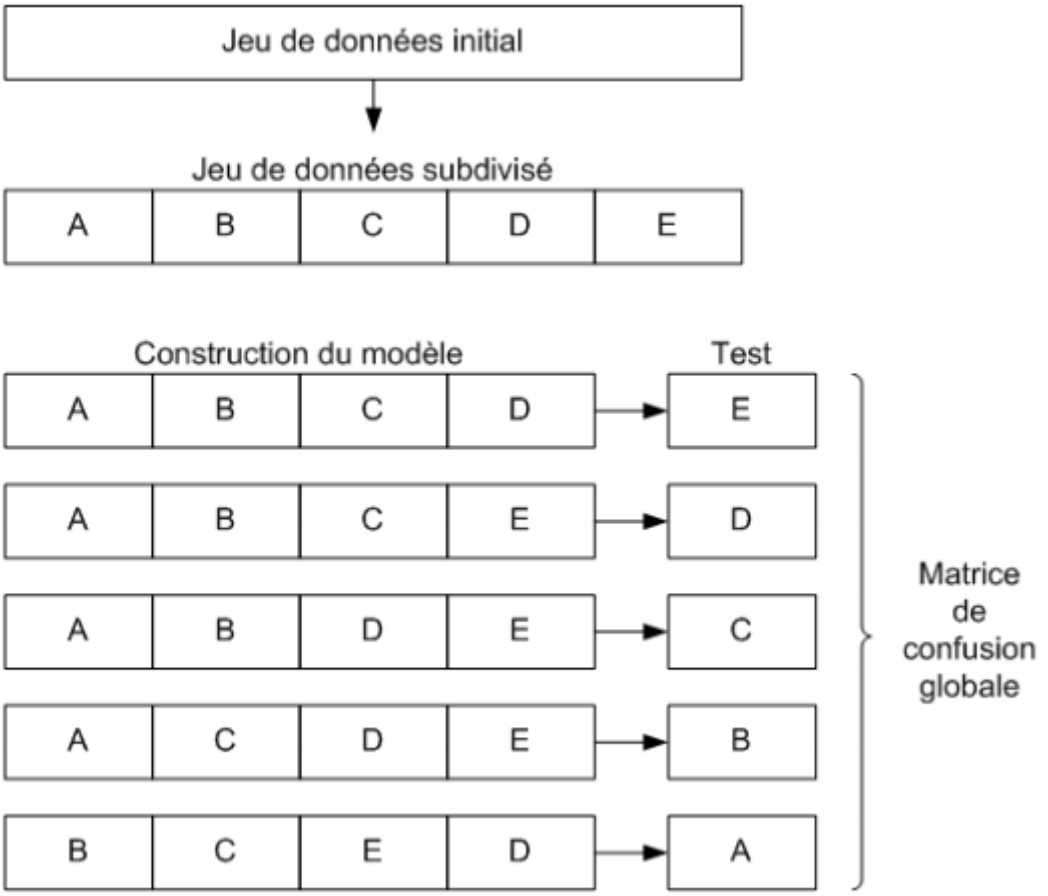


Figure 2.1 k-fold crosss validation en cinq subdivisions [14].

Pour obtenir finalement :

$$j_{ridge}(w) = \sum_{i=1}^{n} (y_i - \hat{Y}_i) + \lambda \sum_{j=1}^{p} {w_j}^2$$

Ce type de régularisation est certes une forme de pénalisation assez efficace mais dans certaines applications le modèle est très complexe on cherche alors à rétrécir en quelques sortes la fonction du modèle de prédiction en d'autres termes certain nombre de paramètres serons nuls. On fait ainsi appel au deuxième type de régularisation qui est le LASSO.

b) La régularisation LASSO

La LASSO est la fonction qui se formule de la manière suivante :

$$R(w) = \lambda \|w\|_1$$

Minimiser les erreurs du modèle revient donc à minimiser la fonction :

$$j_{LASSO(w)} = \sum_{i=1}^{n} (y_i - \hat{Y}_i) + \lambda \|w\|_1$$

Contrairement à la régularisation de norme L2, le LASSO va diminuer la distance entre les solutions possibles, en mesurant n'ont pas la distance euclidienne mais la distance de Manhattan cette fois ci:

$$\|w\|_1 = |w_1| + |w_2| + \cdots + |w_p|$$

La fonction devient ainsi:

$$j_{LASSO(w)} = \sum_{i=1}^{n} (y_i - \hat{Y}_i) + \lambda \sum_{j=1}^{p} |w_p|$$

Remarque : λ est un paramétré de pénalité qui est supérieur à zéro.

c) La régularisation Elastic Net

Pour en bénéficier des deux normes précédentes une combinaison entre LE LASSO et la Régression Ridge était effectuée pour obtenir de meilleurs résultats:

$$j_{Elastic\ Net\ (w)} = \sum_{i=1}^{n} (y_i - \hat{Y}_i) + \lambda\ (\ (1-\alpha)\|w\|_1 + (\alpha)\|w\|_2^2)$$

La régularisation est également appliquée dans le Deep Learning, qui correspond à l'apprentissage des paramètres de connexion d'un réseau de neurones profond (perceptrons multicouche très complexes). Il existe plusieurs méthodes dont on cite :

1. La norme L1et L2 (appliqué dans l'apprentissage profond aussi)
2. Le drop out
3. La data augmentation
4. L'early stopping

2.6 Conclusion

Ce chapitre avait pour objectif la familiarisation avec le Machine Learning en reflétant les différentes applications auquel il est évoqué. De plus, clarifier les méthodes auquel il s’appuie pour construire un modèle de prédiction d’une tâche précis, compte tenu des problèmes dont il fait face lors de sa construction. Dans l’ensemble on catégorise l’apprentissage automatique de supervisé, non supervisé, semi supervisé et apprentissage par renforcement or on se limite dans le supervisé car il fait l’objet de notre étude.

Chapitre 3 Conception et mise en œuvre du système

3.1 Introduction

La COVID-19 est l'un des plus grands défis actuels de santé auquel le monde n'ait jamais été confronté. Notre société est dans une ère de tentatives incroyables pour lutter contre la propagation de cette pandémie mortelle en termes d'infrastructure, de finances, d'affaires, de fabrication, et de plusieurs autres ressources.

Alors que la maladie COVID-19 a déjà infecté plus de 70 000 personnes et décès 1775 vies dans le monde entier au début de mois de février, des études de faisabilité d'application des techniques d'apprentissage automatique ont étés faites pour contrer la COVID-19, mais la plupart étaient encore à leurs débuts, notre modèle en faisait partie.

La conception d'un modèle qui puisse prédire la manière dont le virus pourrait se propager dans le monde, consiste à initialement effectuer une analyse descriptive et explicative des caractéristiques qui définissent l'épidémie auquel on fait face, pour ensuite faire appel aux méthodes du Machine Learning pour la mise en œuvre de cette problématique. Ce chapitre résume ces étapes de conception.

3.2 Méthodologie

Dans cette crise sanitaire actuelle, le Big Data et l'intelligence artificielle se révèlent d'un précieux secours, car si les précautions ne sont pas prises correctement le monde pourrait toujours craindre une nouvelle vague où le risque d'une situation critique est fortement probable. Dans cet esprit construire un système qui aidera à prédire le nombre de cas infectés, dans les 10 prochains jours à l'échelle mondiale n'est qu'un avantage envers l'humanité.

Le système proposé effectue une analyse mathématique pour modéliser cette problématique, en utilisant des algorithmes d'apprentissage automatique (Machine Learning). Selon notre étude bibliographique dans le chapitre précédent, on s'est ramené à utiliser l'algorithme SVR et une Régression linéaire car les deux présentent plusieurs

avances par rapport aux autres algorithmes. La figure 3.1 illustre le schéma de principe du modèle.

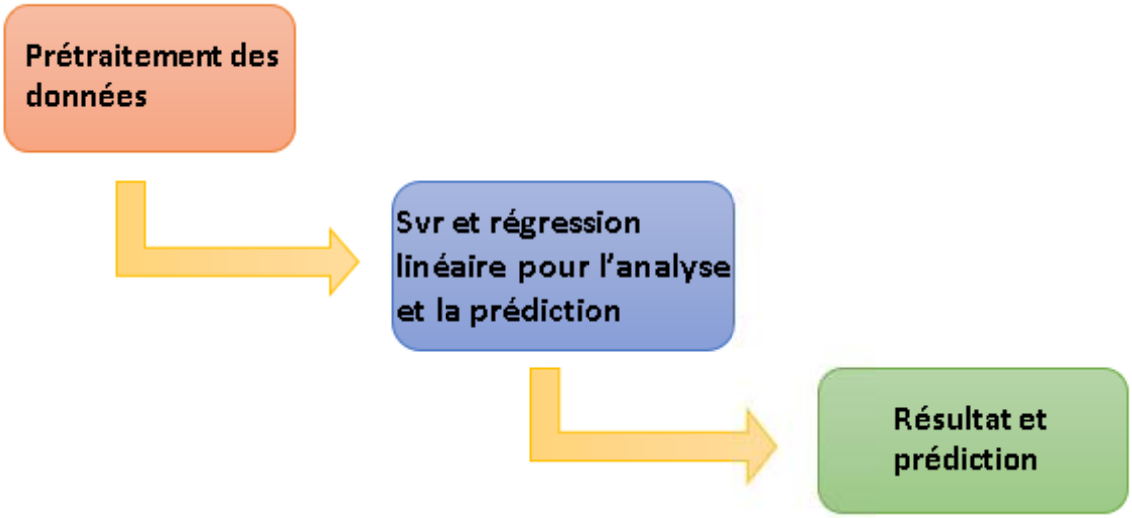


Figure 3.1 Processus du modèle.

3.2.1 Analyse de l'algorithme

a) Régression linéaire

En statistique, la régression linéaire est le schéma permettant de démontrer l'association entre les variables scalaires ou dépendantes et les variables auto-déterminantes solitaires ou supplémentaires (cette section est bien définie dans le chapitre précèdent).

La circonstance d'une seule et unique variable descriptive est appelée régression linéaire simple, tandis que dans le cas de plusieurs variables la procédure différa car on aura à faire à une régression linéaire multiple.

La formulation mathématique de la régression linéaire est comme spécifiée :

$$y = ax + b$$

où x et y sont 2 variables de la droite de régression.

b) Machine à vecteur de support pour régression

La régression vectorielle de support est un choix populaire pour la prédiction et le réglage de courbes pour les types de régressions linéaires et non linéaires. L'algorithme SVR est basé sur les éléments des machines à vecteurs de support (SVM), où les vecteurs de support sont fondamentalement les points les plus proches de l'hyperplan, générés dans un espace

de caractéristiques à n dimensions qui sépare distinctement les points de données autour de ce dernier.

L'équation généralisée pour l'hyperplan peut être représentée par $y = wx + b$, où w est le poids et b est l'intersection à $x = 0$, on définit également la marge de tolérance représentée par epsilon ε.

Le modèle de régression SVR est importé de la bibliothèque python (sklearn) dans la classe des SVM, l'algorithme est ensuite adapté à l'ensemble des données d'entraînement en choisissant les bons paramètres qu'il le définit comme indiqué ci-dessous :

SVR(C=0.01, cache_size=200, coef0=0.0, degree=3, epsilon=0.01, gamma=0.1,
Kernel='poly', max_iter=-1, shrinking=False, tol=0.001, verbose=False)

3.2.2 Méthodes à noyaux

Les méthodes à noyaux ont reçu une attention considérable dans de nombreuses communautés scientifiques, principalement en raison de leur capacité à travailler avec des modèles d'inférence linéaire, permettant en même temps d'identifier les relations non linéaires entre les modèles d'entrées. De plus, la possibilité offerte par ces techniques de travailler de manière homogène avec des données structurées a permis de faire face à différents problèmes tels les problèmes de classification et les problèmes de régression.

La fonction noyau est le composant le plus important des SVR, Il est utilisé pour convertir des données originaires de faible dimension vers un espace de données de plus grande dimension pour ainsi convertir un problème non linéaire en un problème linéaire.

Différentes fonctions du noyau impliquent différentes capacités de mappage, ce qui entraîne par conséquent une précision qui varie selon la fonction choisie. Parmi ces fonctions on cite les plus courantes :

a) Noyau polynomial

Il est populaire dans le traitement d'image, sa fonction est la suivante :

$$k(x_i, x_j) = (x_i . x_j + c)^d$$

où d est le degré du polynôme et c est généralement réglé à 1.

b) Noyau gaussien

L'équation est:

$$k(x_i, x_j) = e^{(-\frac{\|x_i - x_j\|^2}{2\sigma^2})}$$

c) Fonction de base radiale gaussienne (RBF)

C'est un noyau à usage général ; utilisé lorsqu'il n'y a aucune connaissance préalable des données. L'équation est :

$$k(x_i, x_j) = exp(\gamma \parallel x_i - x_j \parallel^2)$$

Pour :

$$\gamma > 0$$

Parfois paramétré avec :

$$\gamma = 1 / 2\sigma^2$$

d) Noyau sigmoïde

Nous pouvons l'utiliser comme proxy pour les réseaux de neurones. L'équation est :

$$k(x_i, x_j) = \tanh(x_i.x_j + c)$$

Remarque :

En raison des informations limitées concernant les fonctions à noyau, les paramètres de cette fonction ne pouvant pas être définie tout.

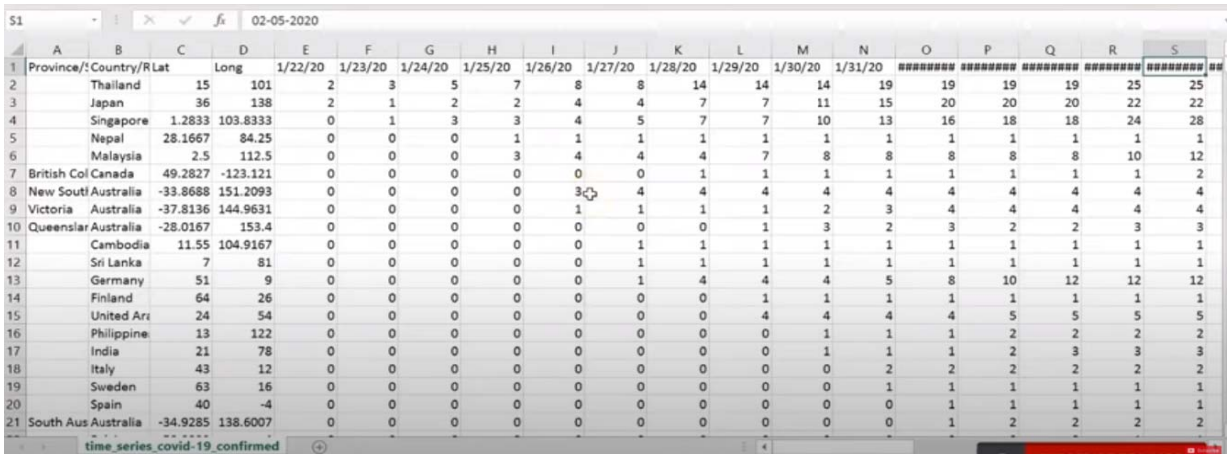
3.2.3 Collecte de données

Alors que l'OMS a déclaré la pandémie comme urgence sanitaire, les chercheurs et les hôpitaux donnent libre accès aux données concernant cette dernière.

Nous avions collecté nos échantillons à partir d'un référentiel de données « kaggel » [1], une communauté en ligne de scientifiques et de praticiens d'apprentissage automatique permettent aux utilisateurs de rechercher et de publier des ensembles de données, et de créer des modèles dans l'environnement du data science basé essentiellement sur le Web.

Environ 10000 données de patients sont stockées montrant des symptômes du virus, ces derniers sont partitionnés dans trois tableaux récapitulatifs des séries chronologiques de rapports des cas confirmés, de décès et des rétablis de la COVID-19 au format CSV. Ceci

est dans 451 pays/régions tel que l'Italie, la chine, l'Amérique, l'inde, etc. La tête du tableau ci- dessous illustre ce qui était introduit.



Province/S	Country/R	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20	1/30/20	1/31/20	########	########	########	########	########
	Thailand	15	101	2	3	5	7	8	8	14	14	14	19	19	19	19	25	25
	Japan	36	138	2	1	2	2	4	4	7	7	11	15	20	20	20	22	22
	Singapore	1.2833	103.8333	0	1	3	3	4	5	7	7	10	13	16	18	18	24	28
	Nepal	28.1667	84.25	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
	Malaysia	2.5	112.5	0	0	0	3	4	4	4	7	8	8	8	8	8	10	12
British Col	Canada	49.2827	-123.121	0	0	0	0	0	0	1	1	1	1	1	1	1	1	2
New Soutl	Australia	-33.8688	151.2093	0	0	0	0	3	4	4	4	4	4	4	4	4	4	4
Victoria	Australia	-37.8136	144.9631	0	0	0	0	1	1	1	1	2	3	4	4	4	4	4
Queenslar	Australia	-28.0167	153.4	0	0	0	0	0	0	0	1	3	2	3	2	2	3	3
	Cambodia	11.55	104.9167	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
	Sri Lanka	7	81	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
	Germany	51	9	0	0	0	0	0	1	4	4	4	5	8	10	12	12	12
	Finland	64	26	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	United Ar	24	54	0	0	0	0	0	0	0	4	4	4	4	5	5	5	5
	Philippine	13	122	0	0	0	0	0	0	0	0	1	1	1	2	2	2	2
	India	21	78	0	0	0	0	0	0	0	0	1	1	1	2	3	3	3
	Italy	43	12	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2
	Sweden	63	16	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
	Spain	40	-4	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
South Aus	Australia	-34.9285	138.6007	0	0	0	0	0	0	0	0	0	0	1	2	2	2	2

Figure 3.2 Tableau csv des cas confirmés.

3.3 Implémentation et langage de programmation

3.3.1 Python

Le langage de programmation Python est apparu à l'époque comme une façon d'automatiser les éléments les plus ennuyeux de l'écriture de scripts, ou de réaliser rapidement des prototypes d'applications. C'est un langage de programmation open source créé par le programmeur Guido van Rossum qui vise à ne pas avoir le besoin d'être compilé pour fonctionner, ceci permet de voir rapidement les résultats d'un changement dans le code. En revanche, il risque d'être plus lent.

Depuis quelques années, ce langage de programmation s'est hissé parmi les langages de programmation les plus utilisés dans le domaine du développement de logiciels, de gestion d'infrastructure et d'analyse de données. Il s'agit d'un élément moteur de l'explosion du Big Data. Par ailleurs, ce langage est très utile dans le domaine de la science des données et du Machine Learning.

Si le Python s'est érigé comme le meilleur langage de programmation pour le Big Data, c'est grâce à ses différents packages et bibliothèques.

3.3.2 Librairies et outils utilisés

a) Outils utilisés

i. Anaconda

Anaconda est une distribution Python pour des traitements intensifs de données, l'analyse prédictive et les calculs scientifiques. Il est fourni de plusieurs bibliothèques dont NumPy matplotlib, pandas, IPython et scikit-learn.

ii. Jupyter Notebook

Jupyter Notebook est un environnement de programmation interactif permettant l'exécution de code et l'insertion d'image et de texte, Il supporte de nombreux langages de programmation, Il est abondamment utilisé dans le data science.

b) Bibliothèques utilisées

i. NumPy

NumPy est un package utilisé pour les calculs scientifiques en Python. Il est idéal pour les opérations liées à l'algèbre linéaire, aux transformations de Fourier, ou aux crunching de nombres aléatoires.

Il peut être utilisé en guise de conteneur multidimensionnel des données génériques. De plus, il s'intègre facilement avec de nombreuses bases de données différentes (dans notre cas les données sont tabulaires). Dans **scikit-learn** les tableaux **NumPy** constituent la structure fondamentale des données.

ii. pandas

Pandas est l'une des bibliothèques de la science de données les plus populaires. Elle a été développée par des Data Scientists habitués au R et au Python, et est aujourd'hui utilisée par un grand nombre de scientifiques et d'analystes.

Elle offre de nombreuses fonctionnalités natives très utiles comme la possibilité de lire des données en provenance de nombreuses sources, de créer des larges data frames à partir de ces sources, et d'effectuer des analyses agrégées basées sur les questions auxquelles on souhaite obtenir des réponses.

Des fonctionnalités de visualisation permettent également de générer des graphiques à partir des résultats d'analyses, ou de les exporter au format Excel. On peut aussi s'en servir pour la manipulation de tableaux numériques et de séries temporelles.

iii. Matplotlib

Le Matplotlib est la bibliothèque de langage de programmation python destinée à tracer et visualiser des données sous forme graphique, elle peut être combinée avec les bibliothèques python comme numpy.

Cette bibliothèque est distribuée librement et gratuitement sous une licence de style BSD, sa version stable actuelle est compatible avec la version 3 du python, Peut être utilisé directement sur Jupyter Notebook.

iv. Random

Random est un module Python regroupant plusieurs fonctions permettant de travailler avec des valeurs aléatoires, ces fonctions peuvent être séparées en trois groupes :

- Celles qui travaillent avec des nombres entiers : telles que random. Randint () et random.randrange () permettent de sélectionner arbitrairement une valeur entière dans un intervalle donné.
- Celles qui travaillent avec des nombres réels : en plus de sélectionner des valeurs dans un intervalle avec les fonctions random.random() et random.uniform() par exemple), permettent aussi de réaliser des distributions gaussienne, exponentielles et logarithmiques.
- Celles qui travaillent avec des séquences (par exemple des listes):permettent de manipuler des éléments dans une liste donnée , elles peuvent sélectionner un élément de la liste aléatoirement (random.choice()), altérer l'ordre des éléments dans la liste elle-même (random.shuffle()) ou encore retourner un nombre d'éléments aléatoires d'une liste (random.sample()).

v. Scikit-learn

Scikit-learn est une bibliothèque dédiée à l'apprentissage automatique, elle est développée par de nombreux contributeurs notamment dans le monde académique par des instituts français d'enseignement supérieurs et de recherche comme ,elle comprend notamment des fonctions pour estimer des forets aléatoires , des régressions logistiques, des algorithmes de classification telles les machines a vecteurs de support (SVM) et comme la bibliothèque de Matplotlib, elle est conçue pour s'harmoniser avec d'autres bibliothèques python, notamment NumPy et SciPy.

3.4 Pourquoi Python ?

Python est devenu le langage de programmation dominant dans le data science, il combine la facilité d'utilisation et d'apprentissage avec la puissance des librairies qu'ils possèdent. C'est un langage multi-paradigme avec de nombreuses implémentations ce qui le rend encore plus intéressant.

Clairement, Python n'est pas vraiment au-dessus des autres langages pour l'intelligence artificielle, mais il s'y prête bien et sa syntaxe concise et facile permet d'y progresser très certainement plus aisément que dans d'autres langages.

3.5 Code sur Python

Dans cette section nous allons voir comment créer un modèle d'apprentissage automatique pour prédire la propagation du virus dans les dix prochains jours à l'aide d'un logiciel python (Jupyter notebook) basé sur la régression linéaire et machine à vecteur de régression (SVR).

3.5.1 Importation des bibliothèques

Commençons maintenant avec le processus de modélisation, la première étape consiste à importer les bibliothèques requises pour la construction du modèle dont Numpy et Pandas. Comme évoqué précédemment, Numpy est une bibliothèque qui contient les fonctions mathématiques nécessaires pour notre problématique, elle est utilisée également pour le calcul scientifique et pour la manipulation des tableaux et matrices à n dimensions. Tandis que Pandas est utilisée pour importer et gérer les ensembles de données. Le code ci-dessous contient les bibliothèques nécessaires pour cette tâche :

```
In [243]: # Importing all the important libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import random
import math
import time
from sklearn.model_selection import RandomizedSearchCV, train_test_split
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, mean_absolute_error
import datetime
import operator
plt.style.use('seaborn')
%matplotlib inline
```

3.5.2 Chargement des données dans le bloc

Après avoir importé tous les bibliothèques requises, on va charger nos données au format csv comme défini précédemment. Un fichier CSV stocke des données tabulaires en texte brut où chaque ligne du fichier est un enregistrement de données.

Nous utilisons la méthode `read_csv`de la bibliothèque pandas pour lire un fichier CSV local en tant que dataframe.

Le code suivant lit les cas confirmés, nombre de décès et le nombre de cas rétablis :

```
[111]: confirmed_cases = pd.read_csv('time_series_covid-19_confirmed.csv')

[112]: deaths_reported = pd.read_csv('time_series_covid-19_deaths.csv')

[113]: recovered_cases = pd.read_csv('time_series_covid-19_recovered.csv')
```

Affichant ensuite les cinq premières lignes à l'aide d'une fonction (head) :

- Pour les cas confirmés :

```
114]: # Display the head of the dataset

      confirmed_cases.head()
```

114]:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	3/6/20	3/7/20	3/8/20	3/9/20
0	NaN	Thailand	15.0000	101.0000	2	3	5	7	8	8	...	48	50	50	50
1	NaN	Japan	36.0000	138.0000	2	1	2	2	4	4	...	420	461	502	511
2	NaN	Singapore	1.2833	103.8333	0	1	3	3	4	5	...	130	138	150	150
3	NaN	Nepal	28.1667	84.2500	0	0	0	1	1	1	...	1	1	1	1
4	NaN	Malaysia	2.5000	112.5000	0	0	0	3	4	4	...	83	93	99	117

5 rows × 58 columns

➢ Pour les décédés :

```
[115]: deaths_reported.head()
```

[115]:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	3/6/20	3/7/20	3/8/20	3/9/20
0	NaN	Thailand	15.0000	101.0000	0	0	0	0	0	0	...	1	1	1	1
1	NaN	Japan	36.0000	138.0000	0	0	0	0	0	0	...	6	6	6	10
2	NaN	Singapore	1.2833	103.8333	0	0	0	0	0	0	...	0	0	0	0
3	NaN	Nepal	28.1667	84.2500	0	0	0	0	0	0	...	0	0	0	0
4	NaN	Malaysia	2.5000	112.5000	0	0	0	0	0	0	...	0	0	0	0

5 rows × 58 columns

➢ Pour les cas rétablis :

```
116]: recovered_cases.head()
```

116]:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	3/6/20	3/7/20	3/8/20	3/9/20
0	NaN	Thailand	15.0000	101.0000	0	0	0	0	2	2	...	31	31	31	31
1	NaN	Japan	36.0000	138.0000	0	0	0	0	1	1	...	46	76	76	76
2	NaN	Singapore	1.2833	103.8333	0	0	0	0	0	0	...	78	78	78	78
3	NaN	Nepal	28.1667	84.2500	0	0	0	0	0	0	...	1	1	1	1
4	NaN	Malaysia	2.5000	112.5000	0	0	0	0	0	0	...	22	23	24	24

5 rows × 58 columns

3.5.3 Prétraitement de données

Avant d'alimenter les données dans le modèle de régression, un prétraitement doit être effectué.

Le prétraitement de données implique la division de ces dernières en attributs et étiquettes puis les diviser en ensembles d'apprentissage et de test.

➢ Pour diviser les données en attributs et étiquettes, on exécute le code suivant :

A l'aide de la fonction key() on peut facilement extraire les clés de nos colonnes (attributs) où chaque élément contient une liste constituent ce dernier (feature extraction) :

```
[117]: # Extracting all the columns using the .keys() function

       cols = confirmed_cases.keys()
       cols
```

```
[117]: Index(['Province/State', 'Country/Region', 'Lat', 'Long', '1/22/20', '1/23/20',
              '1/24/20', '1/25/20', '1/26/20', '1/27/20', '1/28/20', '1/29/20',
              '1/30/20', '1/31/20', '2/1/20', '2/2/20', '2/3/20', '2/4/20', '2/5/20',
              '2/6/20', '2/7/20', '2/8/20', '2/9/20', '2/10/20', '2/11/20', '2/12/20',
              '2/13/20', '2/14/20', '2/15/20', '2/16/20', '2/17/20', '2/18/20',
              '2/19/20', '2/20/20', '2/21/20', '2/22/20', '2/23/20', '2/24/20',
              '2/25/20', '2/26/20', '2/27/20', '2/28/20', '2/29/20', '3/1/20',
              '3/2/20', '3/3/20', '3/4/20', '3/5/20', '3/6/20', '3/7/20', '3/8/20',
              '3/9/20', '3/10/20', '3/11/20', '3/12/20', '3/13/20', '3/14/20',
              '3/15/20'],
             dtype='object')
```

- Gérer et vérifier l'existence des valeurs NaN : La valeur NaN (pas de nombre), est un type de données numériques utilisées pour représenter toute valeur indéfinie ou non représentable. Par exemple, 0/0 n'est pas défini comme un nombre réel et est donc représentée par NaN. Pour cela on veut savoir si nos données contiennent des valeurs nan ou pas, dans la liste unique province commençant par la 1ére valeur par défaut allant jusqu'à la dernière valeur en utilisant la boucle for, si une de ces valeurs est une valeur NaN on les ajoutent à une liste vide « nan_indices » pour qu'on puise ensuite la supprimer des données :« unique_provinces », à l'aide de la fonction pop() comme suit :

```
: # handling nan values if there is any

  nan_indices = []


  for i in range(len(unique_provinces)):
      if type(unique_provinces[i]) == float:
          nan_indices.append(i)

  unique_provinces = list(unique_provinces)
  province_confirmed_cases = list(province_confirmed_cases)

  for i in nan_indices:
      unique_provinces.pop(i)
      province_confirmed_cases.pop(i)
```

- Pour extraire maintenant les colonnes des dates contenant des informations sur les cas confirmés, les décès et les cas guéris (les étiquettes), on emploie la fonction loc qui signifie la spécification des colonnes allant de la quatrième colonne (cols[4]) jusqu'à la dernière colonne (cols[-1]) :

```
[118]: # Extracting only the dates columns that have information of confirmed, deaths and recovered cases

       confirmed = confirmed_cases.loc[:, cols[4]:cols[-1]]

[119]: deaths = deaths_reported.loc[:, cols[4]:cols[-1]]

[120]: recoveries = recovered_cases.loc[:, cols[4]:cols[-1]]

[121]: # Check the head of the outbreak cases

       confirmed.head()
```

[121]:

	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20	1/30/20	1/31/20	...	3/6/20	3/7/20	3/8/20	3/9/20	3/10/20	3/1
0	2	3	5	7	8	8	14	14	14	19	...	48	50	50	50	53	
1	2	1	2	2	4	4	7	7	11	15	...	420	461	502	511	581	
2	0	1	3	3	4	5	7	7	10	13	...	130	138	150	150	160	
3	0	0	0	1	1	1	1	1	1	1	...	1	1	1	1	1	
4	0	0	0	3	4	4	4	7	8	8	...	83	93	99	117	129	

5 rows × 54 columns

➢ Convertir toutes les dates et les cas sous forme d'un tableau numpy array :

```
0]: # Convert all the dates and the cases in the form of a numpy array

    days_since_1_22 = np.array([i for i in range(len(dates))]).reshape(-1, 1)
    world_cases = np.array(world_cases).reshape(-1, 1)
    total_deaths = np.array(total_deaths).reshape(-1, 1)
    total_recovered = np.array(total_recovered).reshape(-1, 1)
```

La bibliothèque scikit-learn, exige qu'un tableau unidimensionnel de variables de sortie (y) soit mise en forme comme un tableau bidimensionnel avec une colonne et des résultats pour chaque ligne.

Pour cela La fonction reshape () est utilisée pour donner une nouvelle forme à un tableau sans changer ses données,et doit être compatible avec la forme d'origine.

Range ()renvoie une séquence de nombres, commençant par 0 et incrémentée de 1 (par défaut), et s'arrête avant un nombre spécifié.

La fonction len() renvoie le nombre d'éléments (ou la longueur) dans un objet.

Finalement, en utilisant la fonction train_test_splitde scikit-learn les données sont divisées en ensembles d'entraînement et de test.Notez que la taille de test est de 0,15 indique que nous avons utilisé 15% de la totale. Pour la sortie train_test_split, nous obtenons x_train_confirmed,x_test_confirmed, y_train_confirmed, et les valeurs y_test_confirmed.

3.5.4 Analyse exploratoire de données

Après avoir soigneusement prétraité notre jeu de données, nous passons à l'étape suivante qui est l'analyse exploratoire des données. Cette étape consiste à comprendre les ensembles de ces derniers en résumant leurs principales caractéristiques.

Il existe des façons pratiquement illimitées d'analyser avec une variété de bibliothèques Python.

D'abord, jetons un coup d'œil sur la manière dont le virus a pu se propager hors la chine durant ces 54 jours en la soustrayant de nos données des cas confirmées.

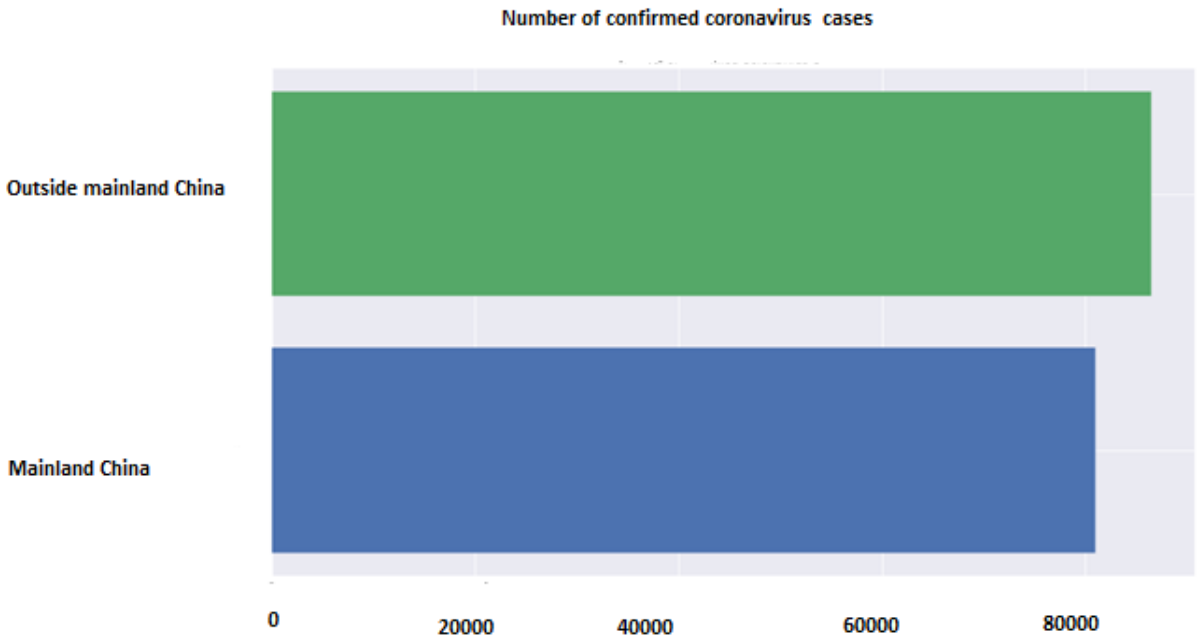


Figure 3.1 Graphique à barres indiquant le total des cas confirmés entre la Chine continentale et en dehors de la Chine continentale.

Dans cette illustration, on remarque clairement que le nombre de cas recensés dans le monde dépasse le compte du territoire chinois, c'est en dehors de la chine que l'épidémie fait rage. Tandis qu'en mi-février, L'épidémie s'y est stabilisée autour de 81.000 cas en ce pays, le reste du monde continue à exploser et dépasse largement les 80000 cas selon la barre du graphe ci-dessus. Pour savoir exactement le nombre de cas infectés en dehors de la chine qui remonte à 86446, il suffit d'utiliser la syntaxe suivante :

```
print('Outside Mainland China {}
cases:'.format(outside_mainland_china_confirmed))
```

Si on veut maintenant visualiser les dix pays les plus touchées par la COVID-19 dans le monde, en regroupant les autres pays dans une catégorie nomée « others », on obtient ce graphique (figure 3.4).

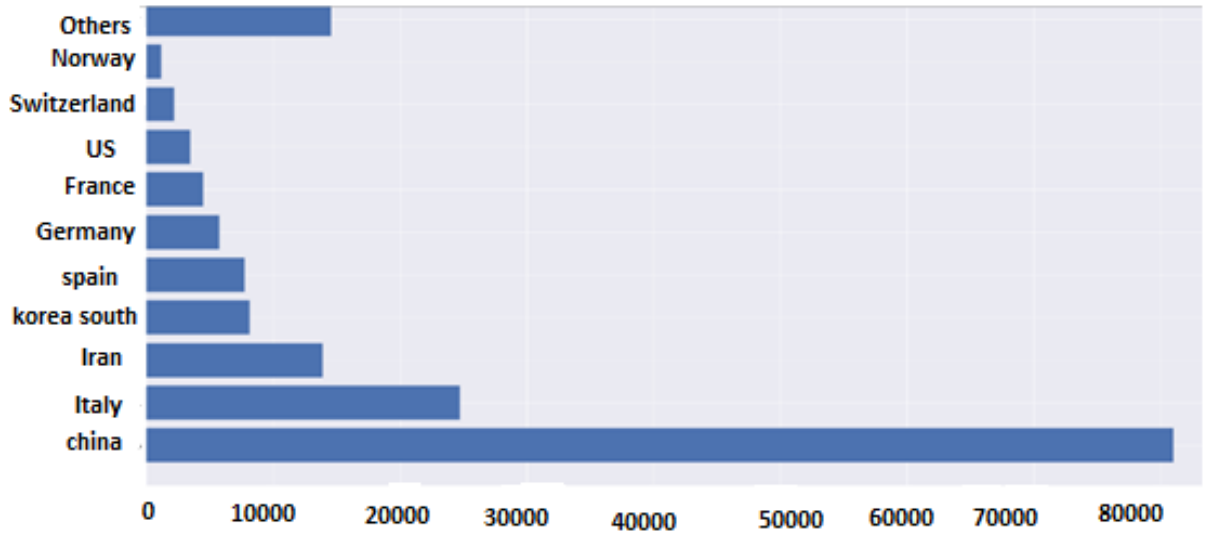


Figure 3.2 Visualisation des 10 pays les plus touchés par la COVID-19.

D'après la figure ci-dessus, la chine est le pays le plus lourdement touché par le virus, 80000 personnes y étaient contaminées, suivi par l'Italie avec plus de 20000 cas, l'Iran 10000 cas, la Corée du sud 8000 cas, visant aussi le continent européen : l'Espagne, l'Allemagne et la France avec presque 5000 cas, notant également la suisse, et la Norvège en s'émergeant finalement aux États-Unis 3000 cas. Les résultats montrent que ces pays suivent la même trajectoire que la chine, c'est-à-dire une croissance incontrôlable et rapide dans le nombre de cas.

Un autre graphe qui correspond cette fois ci au taux de mortalités, c'est le rapport (décès/cas confirmés), il est visualisé dans la figure 3.2. On constate ici que les pays situés dans les rangs supérieurs (au-dessus de la ligne pointillée qui représente la moyenne du rapport définie précédemment) ont le taux de mortalités le plus élevé correspondant aux pays endeuillés par le virus. Tan disque les pays situés en bas de la ligne représentent les zones les moins touchées par la COVID-19, car le taux de mortalités est relativement faible par rapport à la première classe, il est au-dessous de la moyenne. Pour finaliser la visualisation de notre Data set, on illustre une analyse comparative du nombre des cas rétablis et le nombre des décès dans le monde entier :

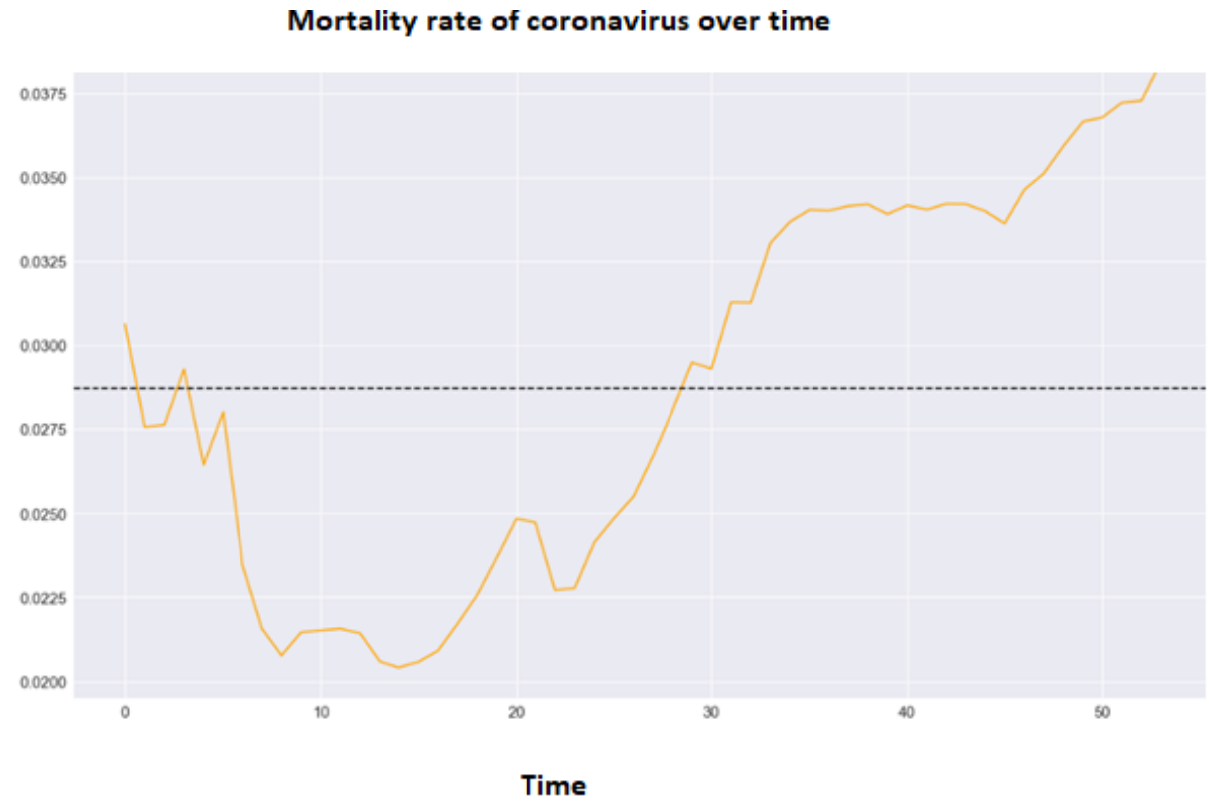


Figure 3.3 Taux de mortalité.

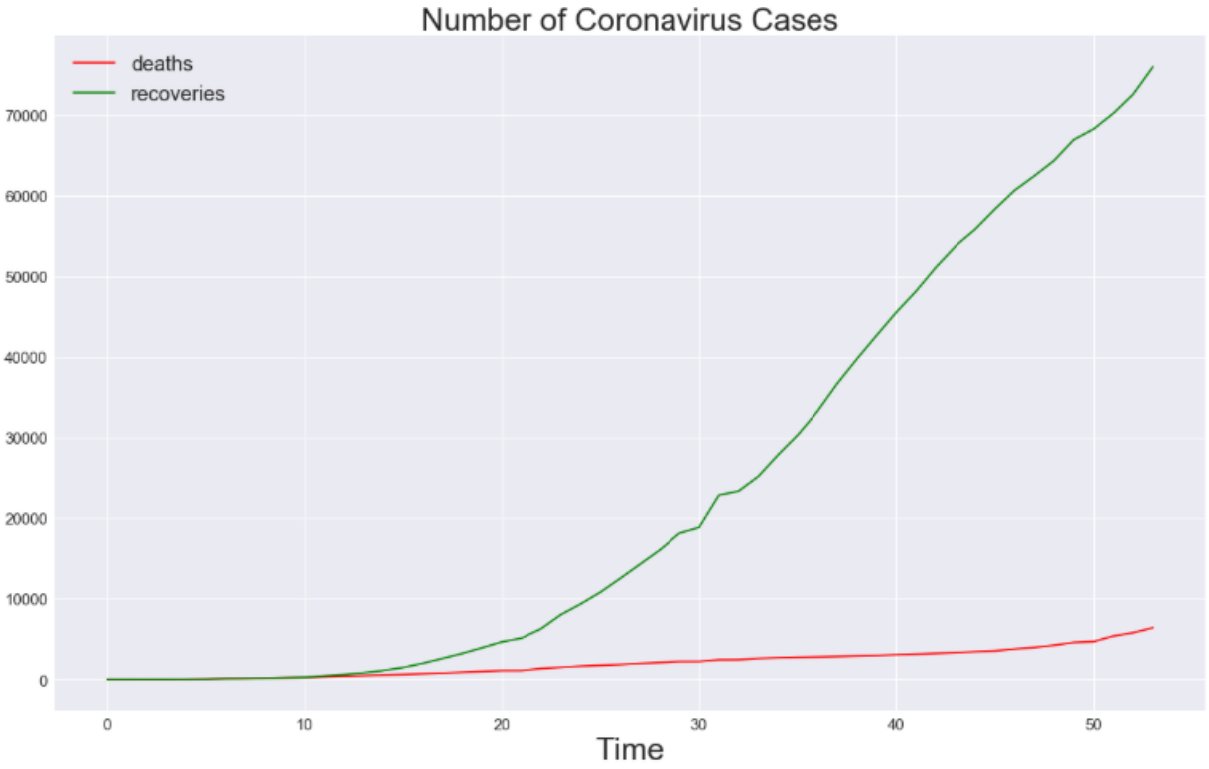


Figure 3.4 Nombre de décès et de cas rétablis.

On remarque clairement dans cette figure que le nombre de décès est presque stable, tandis que le nombre de cas rétablis évolue d'un ordre croissant à un rythme satisfaisant par rapport au temps, en arrivant jusqu'à 70000 cas au début de mars.

3.5.5 Entrainement du modèle pour la prédiction

Après la collecte des données, le prétraitement de ces derniers et l'analyse par graphes et histogrammes vient la dernière étape pour la construction de notre modèle, qui correspond à l'apprentissage par algorithmes.

a) Régression vectorielle de support

Nous allons utiliser `x_train_confirmed` et `y_train_confirmed` obtenus ci-dessus, pour former notre modèle de régression vectorielle de support. Notons que le début de cette cellule décrit les paramètres où la combinaison que nous voulons tester sur le modèle est semblable à la recherche par grille. Pour cela en utilise la fonction `randomizedsearchcv` pour traiter et ajuster nos hyper paramètres.

Le réglage des hyper paramètres vise à trouver des valeurs adaptées, là où les performances du modèle sont les meilleures et le taux d'erreur soit le moins élevé.

On définit l'estimateur comme étant `svm_grid` pour tous les paramètres que nous voulions vérifier, et le nombre des K-folds à 3 pour entraîner le modèle en contournant les données d'apprentissage et en vérifiant le score sur les données de test en utilisant `svm_search.fit()`.

Les paramètres du modèle choisis sont illustrés ci-dessous :

```
svm_search.best_params_
```

```
{'shrinking': False,
 'kernel': 'poly',
 'gamma': 0.1,
 'epsilon': 0.01,
 'C': 0.01}
```

```
svm_confirmed = svm_search.best_estimator_
svm_pred = svm_confirmed.predict(future_forecast)
```

```
svm_confirmed
```

```
SVR(C=0.01, cache_size=200, coef0=0.0, degree=3, epsilon=0.01, gamma=0.1,
    kernel='poly', max_iter=-1, shrinking=False, tol=0.001, verbose=False)
```

b) Régression linéaire

Lorsqu'il s'agit de l'algorithme de régression linéaire on va commencer tout d'abord par importer la bibliothèque requise pour cette tâche :`LinearRegression`, ensuite l'étape suivante consiste à créer le modèle toute en l'adaptant à nos convictions, et ceci à l'aide des données existantes.

L'instruction LinearRegression () crée la variable comme instance, on peut fournir plusieurs paramètres facultatifs pour notre modèle or les paramètres utilisés dans notre cas sont comme suit :

- fit_intercept est un Booléen (par défaut) qui décide de calculer l'interception b (true) ou de la considérer comme égale à zéro (false).
- normaliser est un booléen (par défaut) qui décide de normaliser (true) ou non les variables d'entrées (false).

Entraînons maintenant notre modèle en employant la fonction linear_model.fit.Une fois que notre modèle est bien réalisé, on peut obtenir les résultats pour vérifier si le modèle fonctionne de manière satisfaisante et l'interpréter en calculant l'erreur quadratique (MSE) et l'erreur absolue (MAE).

3.6 Analyse des résultats

3.6.1 Visualisation des résultats

Les résultats sont distingués sous forme d'un tracé à l'aide de plt.plot, une fonction importée de la bibliothèque matplotlib, elles correspondent aux différences entre les valeurs prédites des deux algorithmes utilisés et les vraies valeurs des cas confirmés appartenant aux données de test.

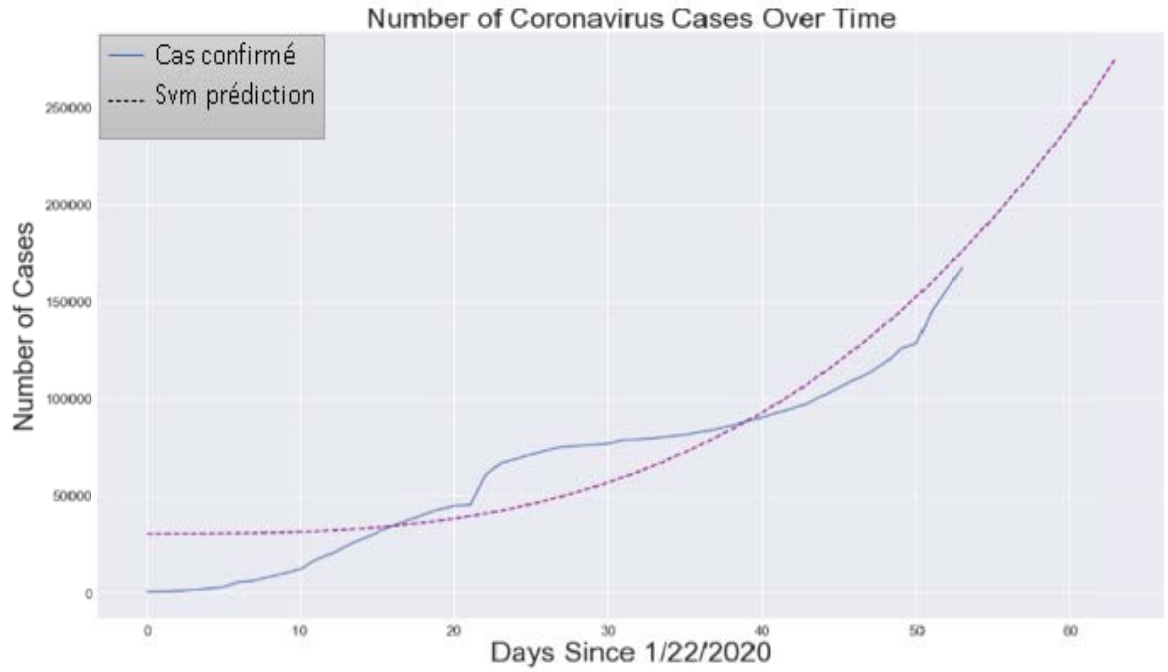


Figure 3.1 Vecteur de régression – nombre de cas.

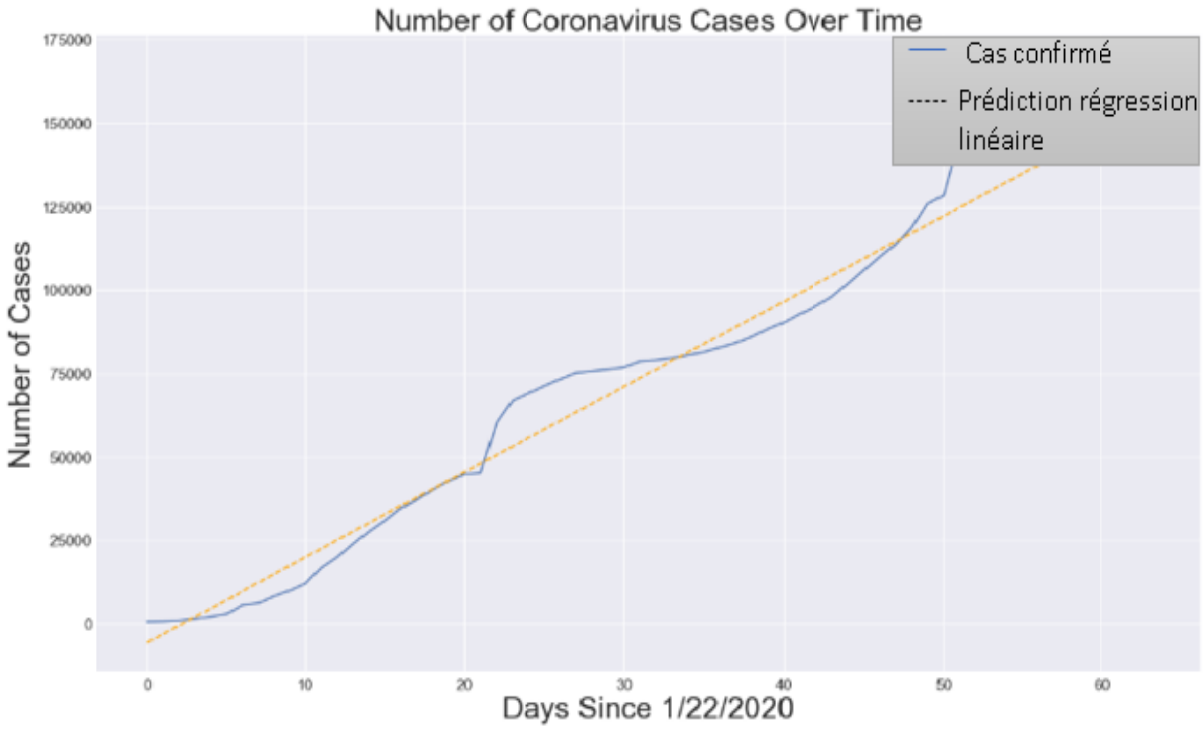


Figure 3.2 Régression linéaire – nombre de cas.

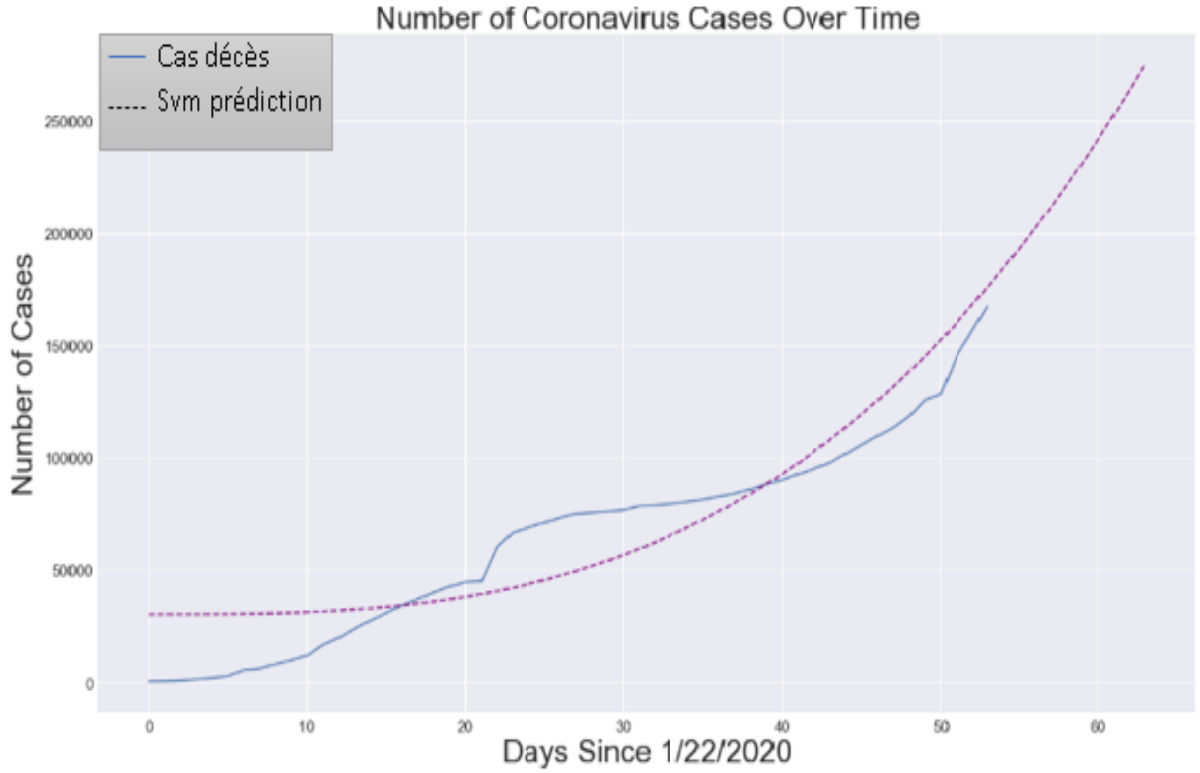


Figure 3.3 Vecteur de régression – nombre de décès.

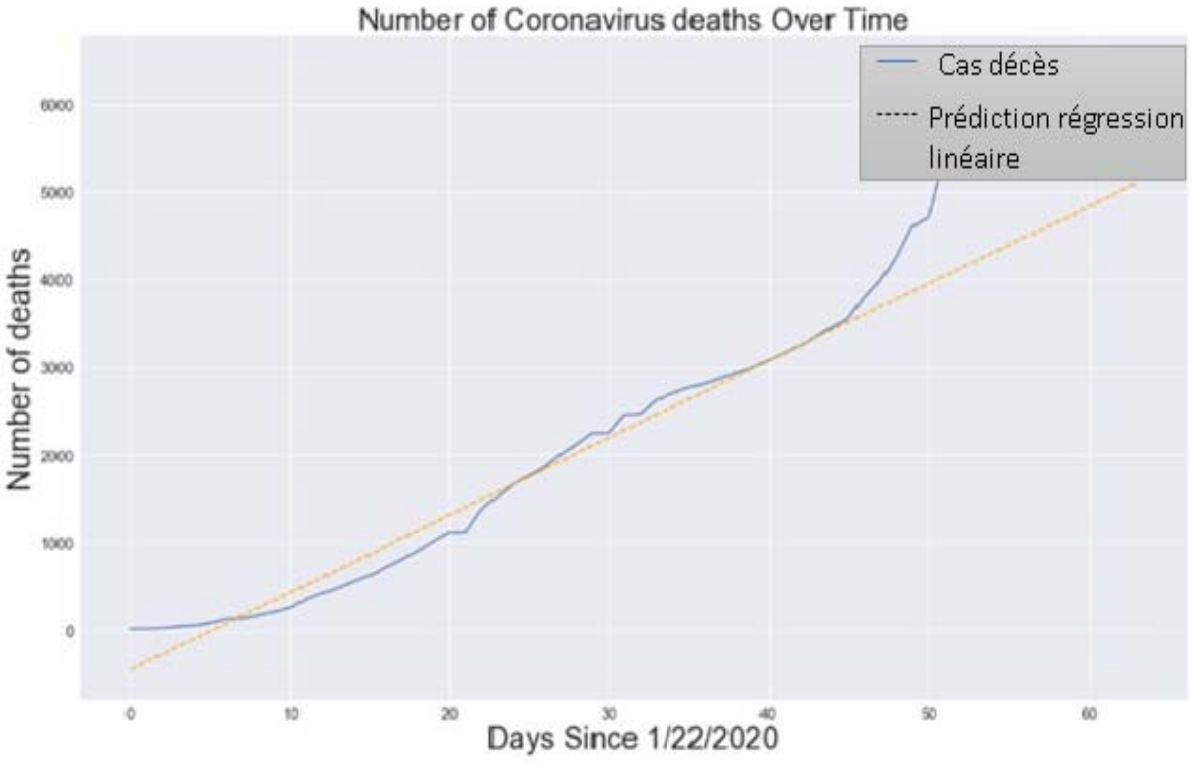


Figure 3.4 Régression linéaire – nombre de décès.

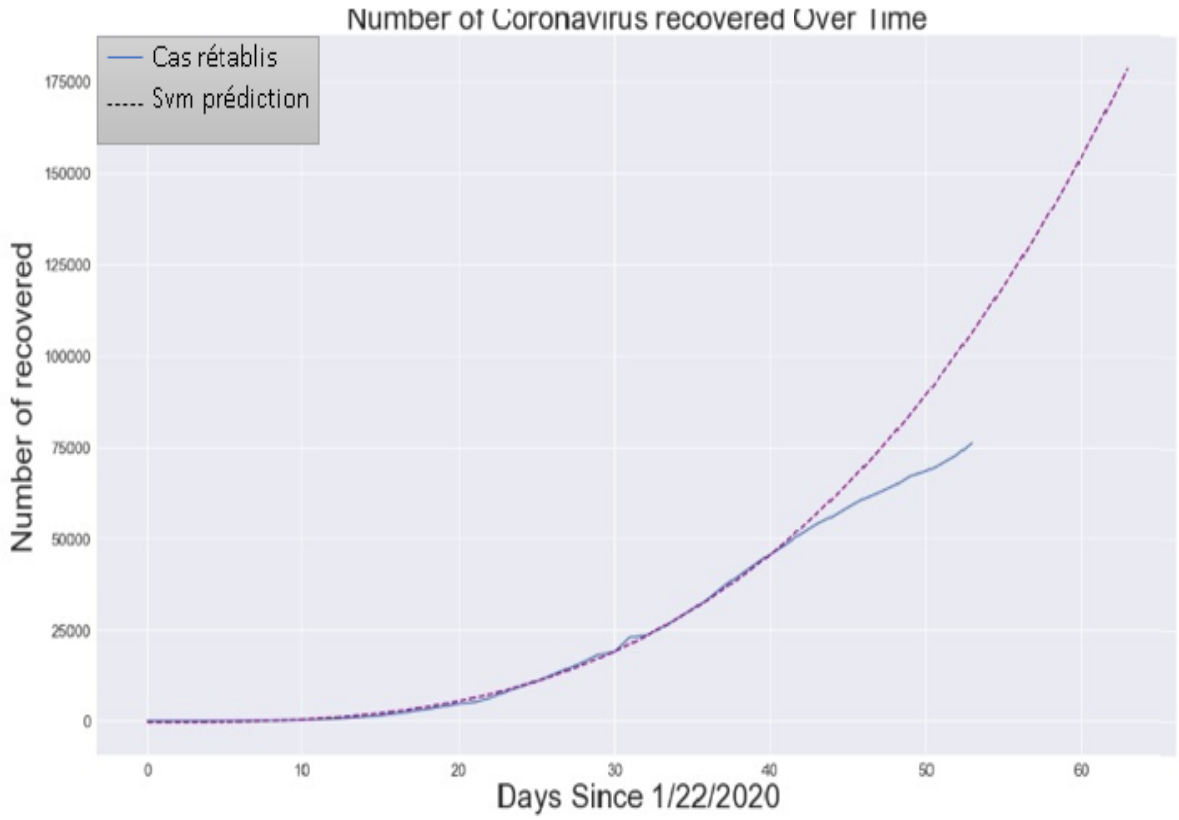


Figure 3.5 Vecteur de régression – nombre de cas rétablis.

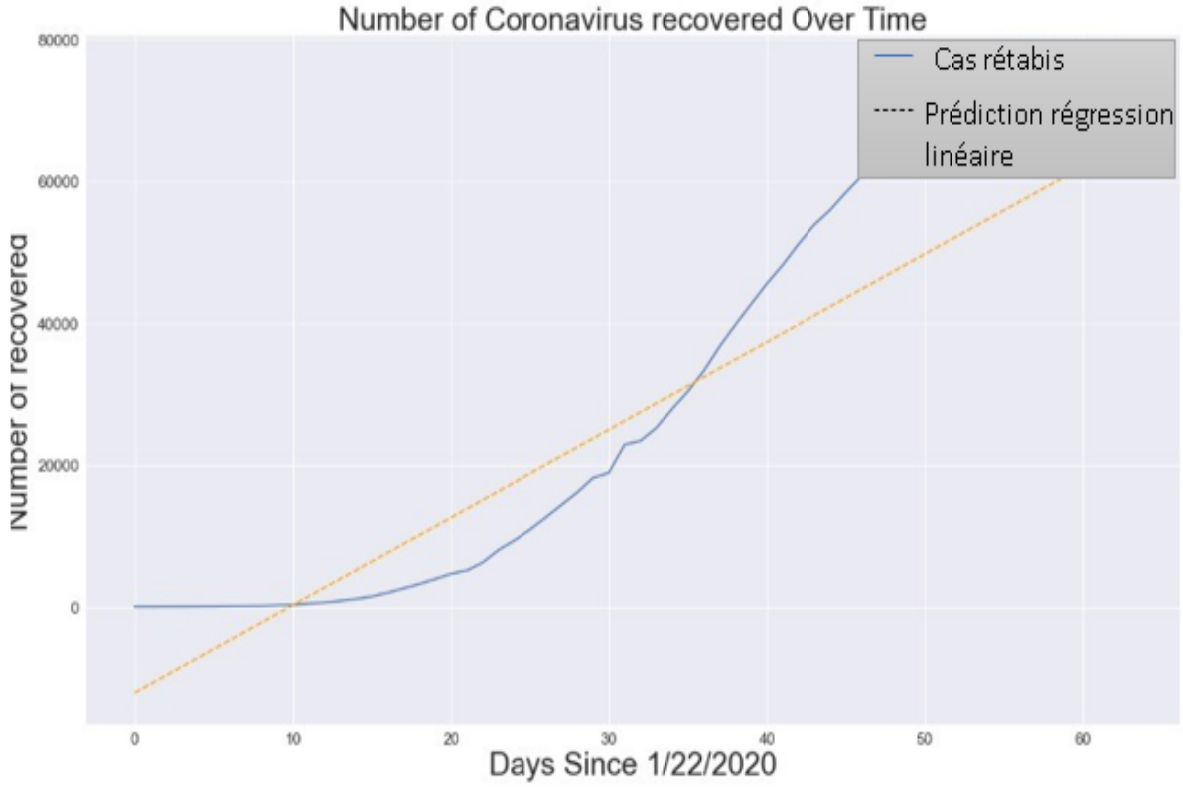


Figure 3.6 Régression linéaire – nombre de cas rétablis.

Lorsque nous voyons la tendance des contaminations au coronavirus dans les deux premières figures, le nombre de cas n'était pas assez élevé au début il était même stable. Au bout du 10^{e} jour l'augmentation a commencé progressivement avec une hausse soudaine en fin janvier, pour continuer à cette trajectoire et s'accélérer de façon éclatante après visuellement on peut estimer que le modèle SVR suit approximativement la trajectoire des vrais cas admis positifs au COVID-19 au contraire du modèle linaire, car au bout du 40^{e} jour le tracé dérive, l'augmentation est constatée exponentielle

En ce qui concerne les cas rétablis, il y a une forte corrélation entre l'évolution des antécédents et ces derniers. En début de pandémie on ne remarque aucun changement positif en raison de la faiblesse des structures sanitaires et la difficulté de faire face à ce virus, mais ceci a changé au bout du 20^{e} jour pour une croissance disant linéaire contrairement aux décès, qui avantageusement malgré l'augmentation rapide des cas le taux de mortalité reste assez faible comparé aux cas contaminés et guéris en atteignant les 6000 cas au 50^{e} jour. Ceci n'est pas totalement rassurent car on suivant le motif sur lequel a évolué l'épidémie du SRAS, ce virus qui est considérablement similaire au COVID-19, la courbe globale est estimée aplatie à la fin où il y'aura plus au moins une diminution

considérable du nombre des cas. Ce n'est pas le cas ici or le pic n'était même pas atteint en cette période-là. Cependant, on peut dire que la trajectoire est en quelque sorte convexe et suit le modèle construit par l'algorithme de régression vectorielle comme était le cas du premier tracé.

3.6.2 Évaluation de performance du modèle

La visualisation ne suffit pas, on doit impérativement évaluer Les paramètres de performance de notre modèle pour vérifier la fiabilité de la prédiction du résultat et s'assurer de nos estimations. Dans le cas d'un problème de régression deux types d'erreurs descriptives sont généralement utilisées :

MSE qui définit l'erreur quadratique moyenne, c'est une métrique de risque correspondant à la valeur attendue d'une perte quadratique :

$$MSE = \frac{\sum_{i=1}^{n}(\widehat{y_i} - y_i)^2}{n}$$

MAE qui définit l'erreur absolue moyenne, une mesure de risque correspondant à la valeur attendue d'une perte absolue :

$$MAE = \frac{\sum_{i=1}^{n}|\widehat{y_i} - y_i|}{n}$$

où $\widehat{y_i}$ présente la valeur prédite d'une observationy_i et i l'étiquette attribuée à cette observation

On définit également Le score qui correspond au pourcentage de précision du modèle :

$$explained - variance(y, \hat{y}) = 1 - \frac{Var\{y - \hat{y}\}}{Var\{y\}}$$

Tableau 3.1 Erreur et précision du modèle pour les cas infectés.

	MSE	MAE	SCORE	Précision
Algorithme SVR	284718501.4	16263.3	0.939	93.9%
Algorithme de régression linéaire	307996364	11965.5	0.923	92.3%

Tableau 3.2 Erreur et précision du modèle pour les cas décédés.

	MSE	MAE	SCORE	Précision
Algorithme SVR	121009.8	317.5	0.819	81.9%
Algorithme de régression linéaire	1230311.8	863	0.423	42.3%

Tableau 3.3 Erreur et précision du modèle pour les cas rétablis.

	MSE	MAE	SCORE	Précision
Algorithme SVR	38354736.5	17906.8	0.646	64.6%
Algorithme de régression linéaire	337147862.9	18220.2	0.825	82.5%

Constations :

Après l'analyse comparative des deux modèles pour chaque catégorie de patients en calculant le score par la manière définit précédemment, on confirme que l'algorithme SVR est plus adapté en tant que modèle de prédiction pour les futurs cas infectés et décédés avec un pourcentage de précision de 93.9% et 81.9% respectivement. Tandis qu'aux cas récupérés, l'approche de régression linéaire est l'optimum avec un pourcentage qui s'élève à 82.5%.

Après ces constations, voici donc nos prédictions pour les 10 prochains jours du 16 au 25 mars pour les cas infectés, les décès et les cas récupérés :

Tableau 3.4 Prédictions pour 10 jours.

Dates	16-mars	17-mars	18-mars	19-mars	20-mars	21-mars	22-mars	23-mars	24-mars	25-mars
Modèle SVR pour les cas infectés	184301.3	193011.4	202044	211405.1	221100.5	231136	241517.6	252251.05	263342.2	274797.02
Modèle SVR pour les cas décédés	6323.41	6641.7	6971.9	7314	7668.4	8035.2	8414.7	8807	9212.4	9631.1
Régression linéaire pour les cas	54648.1	55884.3	57120.5	58356.7	59592.9	60829.2	62065.4	63301.6	64537.85	65774.07

3.6.3 Modèle de prédiction entre les cas prédits les cas réels déclarés par L'OMS

Après la mise en œuvre et l'analyse du modèle, l'exactitude des résultats avec l'environnement en temps réel était nécessaire. Aujourd'hui selon les cas déclarés par L'OMS en période de mars (du 16 au 25), les statistiques du bilan se rapprochait de manière approuvable des résultats prédits par notre modèle construit lorsqu'il s'agissait des cas positifs au COVID-19, car en prenant le 9e jour comme échantillon de notre prédiction, le nombre de cas cumulatifs que le modèle prévoit était de 263342.2 alors que le nombre réel est affirmé à 372755. Une analyse comparative des prédictions est illustrée dans le tableau 3.5.

Cependant, nos estimations qui ressortent des études de modélisation concernant le nombre total de décès et de guérisons étaient un peu déplorables malgré l'optimalité du modèle construit comme illustré dans le tableau 3.5.

Tableau 3.5 Nombre de cas contaminés/décès prédit/réels sur 10 jours.

Dates		16-mars	17-mars	18-mars	19-mars	20-mars	21-mars	22-mars	23-mars	24-mars	25-mars
Modèle SVR	cas contaminés	184301.3	193011.4	202044	211405.1	221100.5	231136	241517.6	252251.05	263342.2	274797.02
	décès	6323.41	6641.7	6971.9	7314	7668.4	8035.2	8414.7	8807	9212.4	9631.1
Les Cas réels selon L'OMS [21]	cas contaminés	167515	179111	191127	209839	234073	266073	292142	332930	372755	413467
	décès	6606	7426	7807	8778	9840	11183	12783	14509	16231	18433

Tableau 3.3 Nombre de cas rétablis prédit/réels sur 10 jours.

Dates	16-mars	17-mars	18-mars	19-mars	20-mars	21-mars	22-mars	23-mars	24-mars	25-mars
Modèle de Régression linéaire	54648.1	55884.3	57120.5	58356.7	59592.9	60829.2	62065.4	63301.6	64537.85	65774.07
Les Cas réels selon L'OMS	77191	79739	82771	84637	88257	91578	95410	98914	104322	108850

Constatations :

Nos prévisions suggèrent une augmentation continue des contaminations tel était le cas en cette période de la pandémie, avec une incertitude considérable associée dans les cas

des décès. On estime que d'autres facteurs rentrent en jeu dont le système immunitaire qui diffère d'une personne à une autre, le risque de réinfection et le disfonctionnement des systèmes de santé au début de cette crise, qui ont fait que les cas réels étaient clairement plus élevés que nos prédictions.

Conclusion générale

Ce projet de fin d'étude traite l'apprentissage automatique dont l'objectif principal était d'étudier un modèle qui puisse prédire les cas cumulatifs de contaminations, de décès, et de guérisons liés à cette pandémie actuelle (COVID-19) et ceci dans le monde entier, en essayent d'avoir une idée sur la propagation du nouveau coronavirus en période du passé et approximer au maximum les nombres résultants au réels cas d'aujourd'hui.

Ce travail été abordé en utilisant des méthodes numériques basées sur des données passées d'une histoire proche. Les prévisions ont été présentées pendant dix jours du 16 au 25 mars 2020, compte tenu du nombre des guérisons et le nombre des décès. Ces données ont aidé à faire face aux défis futurs et à planifier des mesures préventives en termes de distanciation sociale, confinement, port de masque, etc. Qui sont jusqu'à l'heure actuelle un besoin pour réduire le nombre de cas ou de freiner l'arrivé d'une autre vague cruciale.

La réalisation de notre système de prédiction a inclus deux types d'algorithmes de régression robuste, les machines à vecteur de support et la régression linéaire ce qui nous a ramené en outre à une analyse comparative entre ces méthodes déployées dans le système. Les résultats prédictifs étaient satisfaisants en ce qui concernaient les cas positifs au COVID-19, mais du côté des patients récupérés ou décédés y'avait un certain déphasage. Une chose est sure, la propagation de la maladie était significativement élevée et c'est pour cette cause que plusieurs mesures de méfiance appropriées avec une distanciation sociale et une hygiéniste ont été et sont jusqu'à maintenant maintenue.

Bibliographie

[1] https://www.idrc.ca/fr/nouvelles/utiliser-lintelligence-artificielle-et-les-donnees-pour-combattre-la-covid-19-dans-le-sud

[2] https://www.macsf.fr/Responsabilite-professionnelle/Actes-de-soins-et-technique-medicale/intelligence-artificielle-sante#1 .

[3] Anit N Roy, Jais Jose, Aswin S, Neha Gautam, Deepa Nathalia, Arjun Suresh, Prediction and Spread Visualization of Covid-19 Pandemic using Machine Learning, Kerala University of Health Sciences, Thrissur, Amity University, Uttar PradeshUniversity of Malaysia, Haryana Sarasvati Heritage Development Board, may 9 , 2020.

[4] Cem Direkoglu, and Melike Sah, Worldwide and Regional Forecasting of Coronavirus (Covid-19) Spread using a Deep Learning Mode, Middle East Technical University and Near East University, Turkey, may 26 ,2020

[5] Xia Jiang, Richard E. Neapolitan, Artificial Intelligence: With an Introduction to Machine Learning, Second Edition (Chapman & Hall/CRC Artificial Intelligence and Robotics Series),12 mars 2018

[6] Simon James Fong, Jyotismita Chaki, Nilanjan Dey, Artificial Intelligence for Coronavirus Outbreak, springer, 23 juin 2020

[7] https://www.ass-security.fr/blog/camera-thermique-pour-la-detection-du-covid-19-coronavirus/

[8] Stéphane Doncieux, des robots qui apprennent, palais découverte, 2018

[9] https://share.america.gov/fr/lintelligence-artificielle-arme-contre-la-covid-19/

[10] https://information.tv5monde.com/info/covid-19-vaccins-la-securite-avant-tout-374322

[11] http://www.europesolidaire.eu/article.php

[12] Andreas C. Müller, Sarah Guido, ¨ Introduction to Machine Learning with Python ¨, O'Reilly Media, Inc, October 2016.

[13] Chloé-Agathe Azencott, ¨ Introduction au Machine Learning¨ , Dunod, 2019.

[14] Julien Thomas,¨ Apprentissage supervisé de données déséquilibrées par forêt aléatoire¨ ,Autre [cs.OH],Université Lumière Lyon II, 2009.

[15] Laurent Miclet , Antoine Cornuéjols , ¨Apprentissage artificiel - Concepts et algorithmes¨, Eyrolles , 3 juin 2010

[16] Zineb Noumir, Paul Honeine, Cédric Richard.¨ Classification multi-classes au prix d'un classifieur binaire¨. Actes du 23-ème Colloque GRETSI sur le Traitement du Signal et des Images, Bordeaux, France ,2011

[17] Xin-She Yang, ¨Introduction to Algorithms for Data Mining and Machine Learning¨ 1st Edition, Kindle Edition ,Elsevier Science Publishing Co Inc, San Diego, United States ,19 Jun 2019

[18] Quinlan JR.¨ Induction of decision trees¨. Mach Learn. 1986;1(1):81–106.

[19] https://www.saedsayad.com/support_vector_machine_reg.htm

[20] https://www.kaggle.com/vignesh1694/covid19-coronavirus

[21] https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports