

# République algérienne démocratique et populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université 20 Août 1955 SKIKDA



Faculté des sciences

Département d'informatique

Filière : Informatique

Spécialité : Réseaux et systèmes distribués

## Mémoire présenté pour l'obtention du diplôme de Master Académique

Par : BOUSSOUALA Ahcène

LEMRAJJI Azzouz

*Thème*

### *Détection et prédiction des incendies des forêts en utilisant l'intelligence artificielle*

Présenté et soutenu le: *01 juillet 2024* devant le jury composé de:

M <sup>r</sup> . BOUCHARBET Mohamed	Université de Skikda	Président
M <sup>r</sup> . BOULAICHE Mehdi	Université de Skikda	Encadreur
M <sup>me</sup> . RAMDANE Chafika	Université de Skikda	Examineur

Année universitaire : 2023 / 2024



**République algérienne démocratique et populaire**

Ministère de l'enseignement supérieur et de la recherche scientifique

Université 20 Aout 1955 SKIKDA



Faculté des sciences

Département d'informatique

Filière : Informatique

Spécialité : Réseaux et systèmes distribués

**Mémoire présenté pour l'obtention du diplôme de  
Master Académique**

Par : BOUSSOUALA Ahcène

LEMRADJI Azzouz

*Thème*

***Détection et prédiction  
des incendies des forêts  
en utilisant l'intelligence artificielle***

Présenté et soutenu le: *01 juillet 2024* devant le jury composé de:

M <sup>r</sup> . BOUCHARBET Mohamed	Université de Skikda	Président
M <sup>r</sup> . BOULAICHE Mehdi	Université de Skikda	Encadreur
M <sup>me</sup> . RAMDANE Chafika	Université de Skikda	Examineur

Année universitaire : 2023 / 2024

## *Dédicace*

*Je dédie ce travail à :*

*Mes très chers parents que dieu prolonge leur vie,*

*Ma chère épouse ma source d'énergie ;*

*Mes chers trois enfants : Raḥaf, Racil et Abderrahim ;*

*Mes chères sœurs, mes neveux et nièces, chaqu'un par son nom ;*

*Tous la famille et amis ;*

*A tous ce qui m'ont encouragé et aidé durant ces trois dernières  
années d'étude et à réaliser ce modeste travail ;*

*"Il n'y a pas d'ascenseur pour la réussite, il faut prendre l'escalier".*

*Boussouala Ahcène*

## *Dédicace*

*Avec une profonde gratitude et un immense respect, je dédie cet  
humble travail :*

*À ma mère, que Dieu bénisse son âme ;*

*À mon père, que Dieu prolonge sa vie ;*

*À ma chère épouse ;*

*À mes chers enfants : Rawane, Yahia et Abd Errahim.*

*Votre soutien et votre amour inconditionnels ont été ma source  
d'inspiration et de force.*

*Lemradji Azzouz*

## *Remerciements*

*En ce moment empreint d'émotion et de gratitude, nous nous tournons d'abord vers le Dieu Tout-Puissant, source de toute sagesse et de toute bénédiction, pour nous avoir guidés et soutenus tout au long de ce parcours.*

*Nous exprimons également notre profonde reconnaissance envers tous les enseignants qui ont partagé avec nous leur savoir, leur expertise et leur passion. Leur dévouement et leur engagement ont été des phares dans notre quête de connaissance.*

*Nous tenons tout particulièrement à remercier notre encadreur, **Mr BOULLAICHE Mehdi**, dont les conseils avisés, l'écoute attentive et la patience infinie ont été d'une inestimable valeur pour nous. Sa présence bienveillante et son soutien constant ont été des piliers sur lesquels nous avons pu nous appuyer.*

*Enfin, nous adressons nos remerciements à toutes les personnes qui, de près ou de loin, nous ont apporté leur aide, leur soutien et leurs encouragements tout au long de ce parcours. Leur contribution, aussi modeste soit-elle, a été une source de motivation et de réconfort.*

*À chacun d'entre eux, nous disons merci du fond du cœur pour avoir rendu cette aventure enrichissante et mémorable.*

*Avec toute notre gratitude.*

***Boussouala Ahcène***

***Lemradji Azzouz***

# Table des matières

---

Dédicace.....	i
Remerciements .....	iii
Table des matières.....	iv
Liste des figures .....	viii
Liste des tableaux .....	x
Résumé .....	xi
الملخص .....	xii
Abstract .....	xiii
Introduction générale.....	01
Chapitre 1 Les incendies de forêts .....	03
Introduction .....	03
1- Définitions .....	03
1-1 Le feu.....	03
1-2 L'incendie .....	03
1-3 La forêt.....	04
1-4 L'incendie de forêt .....	04
2- Les causes principales des incendies de forêts .....	04
2-1 Les causes naturelles .....	04
2-2 Les causes humaines intentionnelles.....	05
2-3 Les causes humaines accidentelles.....	05
3- Les éléments essentiels pour qu'un feu se produise .....	05
3-1 Combustible .....	05
3-2 Comburant .....	05
3-3 Source de chaleur .....	06
4- Les types des incendies de forêts .....	06
4-1 Feux de profondeur .....	06
4-2 Feux de surface .....	07
4-3 Feux de cime .....	07
5- Les facteurs qui influencent sur la propagation des incendies de forêts .....	08
5-1 Conditions météorologiques .....	08
5-2 Type de combustible .....	08
5-3 La topographie .....	08
5-4 Intensité du feu initial .....	08

# Table des matières

---

6- Les conséquences des incendies de forêts .....	09
7- Comment se prévenir contre les incendies de forêt .....	09
8- Exemples des incendies de forêts .....	10
Conclusion.....	12
Chapitre 2Le concept d'intelligence artificielle .....	13
Introduction .....	13
1- C'est quoi l'intelligence artificielle ?.....	13
- IA Forte .....	13
- IA Faible.....	13
2- Définition de l'IA .....	13
3- Historiques et chronologique de l'IA .....	14
4- L'apprentissage automatique .....	16
5- Le processus de fonctionnement du ML.....	16
6- Les types d'Apprentissage Machine.....	17
6-1- Apprentissage Supervisé .....	18
6-2- Apprentissage Non Supervisé .....	19
6-3- L'apprentissage semi-supervisé.....	20
6-4- Apprentissage par Renforcement .....	20
7- Principaux algorithmes de Machine Learning .....	21
7-1- Régression Linéaire .....	21
7-2- Algorithmes de régression logistique .....	22
7-3- Support Vector Machines (SVM) .....	22
7-4- Arbres de Décision.....	23
7-5- Algorithme K-Means .....	23
7-6- Algorithme K-Nearest Neighbors (KNN) .....	24
7-7- Algorithme Q-Learning .....	24
8- Apprentissage profond (Deep Learning).....	25
8-1- Réseaux de Neurones Convolutifs (CNN) .....	25
8-2- Réseaux de Neurones Récurrents (RNN) .....	26
Conclusion.....	26
Chapitre 3Les datasets et les algorithmes choisis .....	27
Introduction .....	27
1- Description des datasets .....	27
1-1- Dataset pour la prédiction des incendies .....	27

# Table des matières

---

1-2- Dataset pour la détection des incendies .....	30
2- Choix des algorithmes .....	31
2-1- Algorithme K-Nearest Neighbors KNN .....	32
2-1-1- Principe.....	32
➤ La régression .....	33
➤ La classification.....	33
2-1-2- Mesure de distance .....	34
2-1-3- Définir K.....	36
2-1-4- Application de KNN dans l'apprentissage automatique.....	36
2-1-5- Avantages et inconvénients .....	37
2-2- Algorithme Convolutional Neural Network CNN .....	37
2-2-1- Les différentes couches d'un CNN.....	38
2-2-1-1- Couche de convolution .....	38
2-2-1-2- Couche de pooling .....	40
2-2-1-3- Couche de correction ReLU .....	41
2-2-1-4- Couches FullyConnected .....	42
2-2-2- Les avantages d'un CNN .....	42
Conclusion.....	43
Chapitre 4 Implémentation des modèles .....	44
Introduction .....	44
1- Environnement de travail .....	44
1-1- Le langage de programmation Python.....	44
1-2- L'éditeur : Jupyter.....	45
1-3- L'environnement Google Colab.....	45
1-4- Bibliothèques .....	46
1-4-1- TensorFlow .....	46
1-4-2- Scikit-Learn .....	46
1-4-3- NumPy.....	46
1-4-4- Matplotlib .....	47
1-4-5- Keras .....	47
1-4-6- Pandas .....	47
2- Les métriques d'évaluation de modèle de classification.....	47
2-1- Accuracy (Précision Globale) .....	47
2-2- Precision (La précision) .....	48

# Table des matières

---

2-3- Recall (Le rappel) .....	48
2-4- F1 Score .....	48
2-5- ROC-AUC (Receiver Operating Characteristic - Area Under Curve).....	49
2-6- Confusion Matrix (La matrice de confusion) .....	49
2-7- Matrice de corrélation .....	50
<b>3- Implémentation et apprentissage des modèles proposés.....</b>	<b>50</b>
<b>3-1 Le modèle de prédiction Avec KNN .....</b>	<b>51</b>
3-1-1 : chargement du Dataset .....	51
3-1-2 Nettoyage du Dataset .....	51
3-1-3 Conversion des valeurs non numérique .....	52
3-1-4 La matrice de corrélation .....	53
3-1-5 La répartition des classes de feu .....	53
3-1-6 La répartition des causes de feu .....	54
<b>3-2 Le modèle de détection des images de feux Avec CNN.....</b>	<b>55</b>
3-2-1 L'importation du Dataset.....	55
3-2-2 Définir le modèle .....	55
3-2-3 Entraînement du modèle .....	57
3-2-4 La sauvegarde du modèle .....	58
<b>4- Résultats et discussions .....</b>	<b>59</b>
4-1 Le modèle de prédiction des classes de feux par KNN .....	59
4-2 Le modèle de détection des images de feux par CNN .....	62
<b>Conclusion .....</b>	<b>66</b>
<b>Conclusion générale .....</b>	<b>67</b>
<b>Bibliographies.....</b>	<b>68</b>

# Liste des figures

---

## Liste des figures :

Figure 1 : Le triangle du feu .....	06
Figure 2 : Les types des incendies de forêts .....	07
Figure 3 : Les sous domaine de l'IA .....	16
Figure 4 : Apprentissage supervisé .....	18
Figure 5: Apprentissage non supervisé .....	19
Figure 6 : Apprentissage par renforcement .....	20
Figure 7 : Algorithme de régression linéaire .....	21
Figure 8 : Algorithme de régression logistique .....	22
Figure 9: Algorithme support vector machines .....	22
Figure 10: Algorithme arbre de décision .....	23
Figure 11: Algorithme K-Means .....	23
Figure 12: Algorithme K-Nearest Neighbors.....	24
Figure 13 : Réseau de neurones artificiels .....	25
Figure 14 : Extrait de l'ensemble des images .....	31
Figure 15 : Une relation linéaire simple entre deux variables (Exemple de régression) .....	33
Figure 16 : Illustration de KNN avec 03 classes et $K=5$ .....	34
Figure 17 : Schéma du parcours de la fenêtre de filtre sur l'image .....	38
Figure 18 : Pooling avec un filtre $2 \times 2$ et un pas de 2 .....	41
Figure 19: Schéma du fonction d'activation ReLU .....	41
Figure 20 : Logo de l'ID EJupyter .....	45
Figure 21 : Logo de scikit-learn .....	46
Figure 22 : Lignes de code de chargement du Dataset.....	51
Figure 23 : Lignes de code pour le nettoyage du Dataset .....	52
Figure 24 : Lignes de code pour convertir les valeurs non numérique .....	52
Figure 25 : Lignes de code pour tracer la matrice de conrrélation.....	53
Figure 26 : La répartition des classes de feu .....	53
Figure 27 : La répartition des causes de feu .....	54
Figure 28 : Lignes de code pour télécharger le Dataset vers Google colab.....	55
Figure 29 : Lignes de code pour création de modèle CNN .....	56
Figure 30 : Le résultat de l'exécution du code de création du modèle CNN .....	56
Figure 31 : Lignes de code pour entrainer le modèle CNN .....	58
Figure 32 : Ligne de code pour sauvegarder le modèle CNN .....	58
Figure 33 : Ligne de code pour recharger à nouveau le modèle CNN .....	58

## Liste des figures

---

Figure 34 : Evaluation du modèle .....	59
Figure 35 : Les résultats obtenus .....	60
Figure 36 : La matrice de confusion .....	62
Figure 37 : Lignes de code pour calculer les métriques d'évaluation .....	63
Figure 38 : Les résultats de trois métriques .....	64
Figure 39 : Matrice de confusion pour la détection des images .....	65

# Liste des tableaux

---

## Liste des tableaux :

Table 1 : les incendies les plus grands au cours des dernières années .....	11
Table 2 : Les attributs du dataset.....	29
Table 3 : Causes de déclenchement d'un incendie .....	29
Table 4 : les classes des incendies .....	30
Table5 : Exemple de matrice de confusion .....	50

## Résumé

Les incendies de forêts sont devenus une menace environnementale majeure à l'échelle mondiale, exacerbée par le changement climatique et les conditions météorologiques extrêmes. Pour répondre à cette problématique, l'intégration de technologies de pointe telles que l'intelligence artificielle (IA) est devenue cruciale. Ce projet de fin d'études explore l'utilisation de l'IA pour améliorer la détection précoce des incendies de forêt à partir d'images satellite, ainsi que pour prédire et classer leur magnitude. En utilisant des algorithmes avancés de vision par ordinateur et d'apprentissage automatique, notre système vise à surveiller en temps réel de vastes zones forestières, identifiant rapidement les foyers d'incendie et évaluant leur gravité potentielle. Ce travail se concentre spécifiquement sur l'utilisation des algorithmes K-Nearest Neighbors (KNN) pour la prédiction et la classification des incendies de forêt, et des Convolutional Neural Networks (CNN) pour la détection précoce des incendies de forêt. À travers quatre chapitres, nous détaillons l'importance des approches préventives et anticipatives, explorons les principes fondamentaux de l'IA, examinons les ensembles de données utilisés, et présentons le développement et l'évaluation de nos modèles. Ce travail représente une avancée significative dans la gestion des incendies de forêt, offrant des solutions innovantes pour protéger les écosystèmes, les infrastructures et les communautés contre cette menace croissante.

**Mots clé :** Incendies de forêts, Intelligence artificielle (IA), Détection précoce, prédiction, classification des incendies, algorithme KNN, algorithme CNN, Vision par ordinateur, Apprentissage automatique.

## ملخص

يشكل التزايد المستمر في حرائق الغابات تهديداً بيئياً رئيسياً على نطاق عالمي، مما يؤدي إلى خسائر كبيرة من حيث الأرواح البشرية والتنوع البيولوجي والمورد الطبيعية. ولمواجهة هذا التحدي، يصبح استخدام التقنيات الحديثة المتقدمة أمراً حتمياً للتنبؤ والتخفيف من النتائج المدمرة لهاته الكوارث. في هذه الدراسة قمنا بتسليط الضوء على الأهمية الكبيرة لتطوير أنظمة قائمة على الذكاء الاصطناعي لإدارة حرائق الغابات، مع التركيز على الكشف المبكر والتنبؤ وتصنيف الحرائق حسب خطورتها المتوقعة من خلال استغلال رؤية الحاسوب وخوارزميات التعلم الآلي، إذ أن الهدف المرجو الوصول إليه هو مراقبة المناطق الغابية الواسعة في الوقت الحقيقي، وتحديد نقاط اندلاع الحرائق بسرعة والتنبؤ بتطورها، حيث تمكن مثل هاته الأنظمة من الإستجابة بشكل أسرع للحرائق الناشئة وتحديد أولويات التدخل بناء على الخطورة المحتملة.

يرتكز هذا العمل بشكل خاص على استخدام خوارزميات K-Nearest Neighbors (KNN) لتوقع وتصنيف حرائق الغابات، وشبكات التعلم العصبي التكراري (CNN) للكشف المبكر عن الحرائق. من خلال أربع فصول، نوضح أهمية النهج الوقائي والتوقعي، ونتناول المبادئ الأساسية للذكاء الاصطناعي، ونفحص مجموعات البيانات المستخدمة، ونقدم تطوير وتقييم نماذجنا، حيث يمثل هذا البحث تقدماً كبيراً في إدارة حرائق الغابات، مقدماً حلاً مبتكرة لحماية النظم البيئية والبنية التحتية والمجتمعات من هذا التهديد المتزايد، مما يحسن من استغلال للموارد المادية والبشرية المحدودة بطريقة فعالة.

**كلمات مفتاحية:** حرائق الغابات، الذكاء الاصطناعي، الكشف المبكر، التوقع، تصنيف الحرائق، خوارزمية KNN، خوارزمية CNN، رؤية الحاسوب، التعلم الآلي.

## Abstract

Forest fires have become a major global environmental threat, exacerbated by climate change and extreme weather conditions. To address this issue, the integration of advanced technologies such as artificial intelligence (AI) has become crucial. This thesis project explores the use of AI to enhance early detection of forest fires from satellite images, as well as to predict and classify their magnitude. By leveraging advanced computer vision and machine learning algorithms, our system aims to monitor vast forested areas in real-time, swiftly identifying fire outbreaks and assessing their potential severity. This work specifically focuses on the use of K-Nearest Neighbors (KNN) algorithms for predicting and classifying forest fires, and Convolutional Neural Networks (CNN) for early fire detection. Across four chapters, we detail the importance of preventive and anticipatory approaches, delve into the fundamental principles of AI, examine the datasets used, and present the development and evaluation of our models. This research represents a significant advancement in forest fire management, offering innovative solutions to protect ecosystems, infrastructure, and communities from this escalating threat.

**Keywords :** Forest fires, Artificial Intelligence (AI), Early detection, prediction, fire classification, KNN algorithm, CNN algorithm, Computer vision, Machine learning.

# **Introduction générale**



# Introduction générale

---

## Introduction générale

La fréquence croissante des incendies de forêts constitue une menace environnementale majeure à l'échelle mondiale, entraînant des pertes considérables en termes de vies humaines, de biodiversité et de ressources naturelles. Face à cette problématique, l'utilisation de technologies de pointe devient impérative pour anticiper et atténuer les conséquences dévastatrices de ces catastrophes.

L'importance de développer des systèmes basés sur l'intelligence artificielle (IA) pour la gestion des incendies de forêt ne cesse de croître face aux défis posés par le changement climatique et l'augmentation des événements extrêmes. La détection précoce des incendies de forêt grâce à l'exploitation des images satellite, ainsi que la prédiction et la classification de leur magnitude, sont des domaines où l'IA peut jouer un rôle déterminant. En intégrant des technologies avancées de vision par ordinateur et d'apprentissage automatique, il devient possible de surveiller de vastes étendues de terrain en temps réel, d'identifier les foyers d'incendie dès leur apparition, et de prédire leur évolution. Ces systèmes permettent non seulement de réagir plus rapidement aux incendies naissants, mais aussi de prioriser les interventions en fonction de la gravité potentielle des incendies, optimisant ainsi l'utilisation des ressources limitées. En combinant détection et prédiction, l'IA offre des outils puissants pour protéger les écosystèmes, les infrastructures et les communautés, tout en minimisant les impacts économiques et environnementaux des incendies de forêt.

Donc, la question qui se pose est : ***‘Comment l'intelligence artificielle peut-elle être utilisée efficacement pour améliorer la détection précoce des incendies de forêt à partir des images satellite et pour prédire et classer leur magnitude, afin d'optimiser les interventions et de minimiser les impacts environnementaux et économiques?’***

Pour répondre à cette problématique, notre projet de fin d'études propose de développer un système basé sur l'intelligence artificielle, capable d'améliorer la détection précoce des incendies de forêt à partir d'images satellite, ainsi que de prédire et classer leur magnitude. En exploitant des algorithmes de vision par ordinateur et d'apprentissage automatique, nous visons à créer un outil performant pour surveiller en temps réel les vastes étendues forestières, identifier rapidement les foyers d'incendie, et évaluer leur gravité potentielle.

# Introduction générale

---

Notre projet est subdivisé en quatre chapitres :

Le premier chapitre de notre étude plongera dans la réalité des feux de forêts, débutant par une définition précise de ce phénomène destructeur. Nous explorerons les dégâts considérables infligés aux écosystèmes forestiers, à la biodiversité et aux populations humaines. Des statistiques éclaireront l'ampleur croissante de cette menace, soulignant l'impératif d'adopter des approches préventives et anticipatives.

Le deuxième chapitre sera consacré à l'intelligence artificielle, mettant en lumière ses principes fondamentaux et examinant les différents algorithmes qui la sous-tendent. Nous explorerons l'importance de l'apprentissage machine, du traitement d'image et de l'analyse de données dans le contexte de la prévention des incendies de forêts, jetant ainsi les bases nécessaires pour le développement de notre système de prédiction et détection.

Le troisième chapitre abordera les ensembles de données utilisés dans notre étude, en détaillant leur source, leur prétraitement et leur pertinence pour la détection des incendies de forêt. De plus, ce chapitre se penchera sur le choix judicieux des algorithmes sélectionnés pour la détection, la prédiction et la classification des incendies. Nous justifierons ces choix en fonction de leurs pertinences et de leurs efficacités dans le contexte spécifique des incendies de forêt. Une analyse approfondie des algorithmes retenus permettra d'établir leur validité et leur adéquation pour atteindre nos objectifs.

Le quatrième chapitre marquera l'aboutissement de notre travail par la création et l'apprentissage des modèles de prédiction et détection des incendies. Nous détaillerons les étapes de développement, les paramétrages spécifiques et les choix techniques effectués. Ce chapitre abordera également l'évaluation des résultats obtenus, en analysant la performance et la précision des modèles. L'objectif final est de parvenir à un système opérationnel capable de prédire et détecter les incendies de forêt de manière précoce et précise.

# **Chapitre 1**

## **Les incendies de forêts**



## Introduction

Le feu, un phénomène naturel et contrôlé depuis des millénaires, revêt une importance cruciale dans divers aspects de la vie humaine. Cependant, lorsque le feu échappe à tout contrôle, il peut se transformer en un redoutable incendie de forêt, engendrant des conséquences dévastatrices sur l'environnement et les communautés humaines. Dans ce chapitre, nous explorerons les définitions fondamentales du feu, de l'incendie et de la forêt, ainsi que les divers types d'incendies de forêt et les facteurs influençant leur propagation. Nous examinerons également les conséquences dévastatrices de ces incendies sur la biodiversité, le climat, la qualité de l'air, l'économie et les sols. Enfin, nous aborderons des stratégies essentielles de prévention et de gestion des incendies de forêt pour minimiser les risques et préserver nos écosystèmes.

## 1- Définitions

### 1-1 Le feu

- Le feu est le résultat d'une réaction chimique exothermique appelée combustion, caractérisée par la libération de chaleur, de lumière, de gaz et de vapeur. Il nécessite généralement un combustible, un agent oxydant (comme l'oxygène atmosphérique) et une source de chaleur pour maintenir la réaction.[1]
- Le feu est un processus de combustion caractérisé par la réaction chimique entre un combustible et un agent oxydant, généralement accompagné de la libération de chaleur, de lumière et de gaz. Il représente un phénomène naturel ou contrôlé, essentiel à diverses activités humaines, tout en pouvant être destructeur lorsqu'il échappe à tout contrôle.[2]

### 1-2 L'incendie

Un incendie est un événement caractérisé par la propagation incontrôlée du feu, souvent accompagnée de flammes et de fumée. Il peut résulter de diverses causes, telles que des accidents, des négligences, des actes criminels, ou des phénomènes naturels. Les incendies peuvent causer des dommages matériels, des pertes humaines, et avoir des impacts écologiques significatifs [3]

## 1-3 La forêt

Une forêt est un écosystème terrestre caractérisé par une concentration importante d'arbres, de plantes ligneuses, et d'une diversité biologique associée. Elle constitue un habitat complexe abritant une variété de flore et de faune, et joue un rôle crucial dans la régulation climatique, la conservation de la biodiversité, et la fourniture de ressources naturelles telles que le bois, les fruits et les services éco-systémiques. [4]

## 1-4 L'incendie de forêt

Un incendie de forêt est un feu qui se propage de manière incontrôlée à travers une zone boisée, engendrant une combustion rapide et intense de la végétation. Ces incendies peuvent être déclenchés par diverses causes, y compris la foudre, l'activité humaine, ou des conditions météorologiques extrêmes. Les incendies de forêt ont le potentiel de causer des dommages écologiques étendus, de menacer la vie sauvage, de détruire des habitats naturels et de mettre en danger les communautés humaines à proximité.[5]

## 2- Les causes principales des incendies de forêts

Les incendies de forêt peuvent être déclenchés par diverses causes, qui sont souvent classées en trois catégories principales : les causes naturelles, les causes humaines intentionnelles, et les causes humaines accidentelles. Voici une explication de chacune de ces catégories :

### 2-1 Les causes naturelles

- **Foudre** : Les éclairs peuvent frapper les arbres ou la végétation sèche, déclenchant ainsi des incendies de forêt. Les incendies dus à la foudre sont souvent appelés incendies de foudre.
- **Le changement climatique** : Le changement climatique est un facteur majeur de l'augmentation des incendies. Les vagues de chaleur extrême sont maintenant plus fréquentes qu'il y a 50 ans,[6] et cette tendance devrait s'aggraver avec le réchauffement continu de la planète. Les températures plus élevées entraînent un assèchement du paysage, créant des conditions propices à la multiplication des incendies de forêt, ce qui entraîne une augmentation des émissions de gaz à effet de serre, aggrave les changements climatiques et intensifie les incendies, formant ainsi une boucle de rétroaction climatique. Cette boucle, combinée à l'expansion des

activités humaines telles que l'agriculture dans les régions forestières, est largement responsable de la hausse actuelle de l'activité incendiaire

- **Conditions météorologiques extrêmes** : Des conditions météorologiques extrêmement chaudes et sèches peuvent créer un environnement idéal à la propagation rapide des incendies de forêt.

## 2-2 Les causes humaines intentionnelles

- **Incendie criminelle** : Certains incendies de forêt sont délibérément allumés par des individus dans un acte criminel. Ces incendies peuvent être motivés par diverses raisons, y compris la vengeance, la malveillance, ou d'autres motivations criminelles.

## 2-3 Les causes humaines accidentelles

- **Brûlis agricoles incontrôlés** : Certains agriculteurs utilisent le feu pour nettoyer les terres, mais si ces brûlis ne sont pas correctement contrôlés, ils peuvent se transformer en incendies de forêt ;
- **Feux de camps non maîtrisés** : Des feux de camp mal éteints ou non surveillés peuvent déclencher des incendies dans les zones boisées ;
- Les mégots de cigarettes jetés négligemment ;
- Des étincelles provenant d'équipements mécaniques, etc.

Il est important de souligner que souvent, les incendies de forêt résultent d'une combinaison de plusieurs de ces facteurs. La sensibilisation, la prévention et la gestion appropriée des incendies de forêt sont essentielles pour minimiser les risques et les conséquences de ces événements.

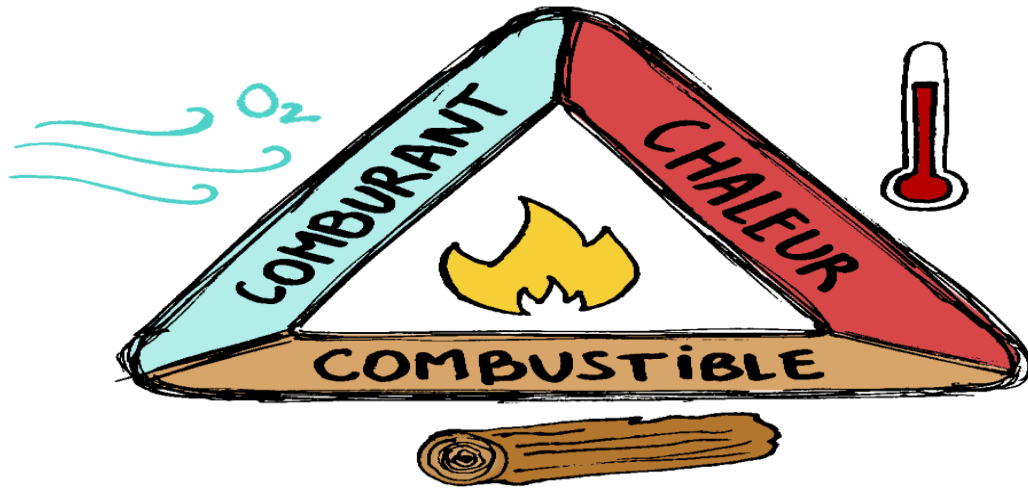
## 3- Les éléments essentiels pour qu'un feu se produise

Le triangle du feu est un concept fondamental en sécurité incendie qui représente les trois éléments essentiels nécessaires pour qu'un feu se produise et persiste. Ces éléments sont souvent représentés sous forme de triangle pour illustrer leur interdépendance. Les trois côtés du triangle du feu sont généralement les suivants :

**3-1 Combustible** : Il s'agit de la substance qui peut brûler. Cela peut être du bois, du papier, de l'essence, des matières plastiques, etc. Le combustible est la matière qui libère de l'énergie lorsqu'elle réagit avec l'oxygène et produit un feu.

**3-2 Comburant** : Le Comburant est l'autre réactif du triangle et de la réaction chimique du feu : la plupart du temps, il s'agit de l'air ambiant et plus particulièrement de l'un de ses composants : le dioxygène

**3-3 Source de chaleur** : Il s'agit de l'énergie initiale nécessaire pour démarrer le processus de combustion. Cela peut être une étincelle, une flamme nue, de la chaleur radiante, etc.



*Figure 1 : le triangle du feu*

Le triangle du feu montre que si l'un de ces trois éléments est éliminé, le feu ne peut pas se produire ou, s'il est déjà en cours, il peut être éteint en retirant l'un de ces éléments. Ce concept est essentiel pour la compréhension de la prévention des incendies et des techniques d'extinction. Certains modèles étendus incluent un quatrième élément, la réaction en chaîne, qui représente la continuation de la combustion une fois qu'elle a commencé. Cependant, le triangle du feu de base reste un outil pédagogique important dans le domaine de la sécurité incendie.

## 4- Les types des incendies de forêts

Il existe trois types d'incendies de forêt : en profondeur, en surface et dans la cime.[7]

### 4-1 Feux de profondeur

Un feu de profondeur (souterrain) brûle dans la matière organique sous la litière de la forêt. Ce type d'incendie est généralement décrit comme un feu couvant ou rampant. La profondeur à laquelle ces feux brûlent dépend de la profondeur des matières organiques et des conditions de sécheresse du moment. Les feux de profondeur peuvent être difficiles à éteindre, car leur contour est souvent difficile à localiser. Les feux qui brûlent en profondeur mettront également plus de temps à s'éteindre.

## 4-2 Feux de surface

Un feu de surface brûle dans la couche de combustible de surface qui comprend la litière supérieure du sol de la forêt, comme les feuilles mortes, les aiguilles et la végétation. Le feu n'atteint pas la cime des arbres. Bien que des flambées en chandelle et du transport de tisons puissent se produire occasionnellement, la propagation du feu demeure près de la surface du sol forestier et ne brûle pas les arbres. La flambée en chandelle est l'embrasement d'un arbre ou d'un petit groupe d'arbres sans la présence d'un feu de cime soutenu, tandis que les transports de tisons sont des étincelles ou des braises portées par le vent et qui déclenchent de nouveaux incendies.

- Les feux de surface qui se déplacent lentement sont décrits comme des feux **rampants**.
- Les feux de surface se déplaçant plus rapidement sont généralement décrits comme des feux **roulants**.

Les feux de surface peuvent se transformer en feux de cime dans les conditions suivantes :

- combustibles continus, de la couche superficielle aux cimes ;
- présence de conifères ;
- vent assez fort pour faire passer le feu à travers les cimes.

## 4-3 Feux de cime

Un feu de cime progresse à travers la couche de combustible de la cime des arbres, le plus souvent simultanément à un feu de surface. Le type, le volume et la disposition des combustibles, de la surface à la cime, sont autant de facteurs qui contribuent au développement des feux de cime. Le feu progresse en un mur de flammes bien défini, qui s'élève de la couche de combustible de surface jusqu'au-dessus de la cime. Les feux de cime se propagent rapidement et entravent considérablement le travail visant à stopper la propagation ou à éteindre l'incendie.

### Feu de profondeur



### Feu de surface



### Feu de cime



*figure2 : Les types des incendies de forêts*

La gestion des incendies de forêt implique souvent une compréhension approfondie de ces différents types d'incendies pour élaborer des stratégies d'extinction appropriées.

## **5- Les facteurs qui influencent sur la propagation des incendies de forêts**

La propagation des incendies de forêt dépend de plusieurs facteurs, notamment les conditions météorologiques, le combustible disponible et la topographie. Voici les principaux facteurs qui influent sur la propagation des incendies de forêt :

### **5-1 Conditions météorologiques**

Les conditions météorologiques jouent un rôle déterminant dans la propagation des incendies de forêt. Les vents forts peuvent transporter les flammes et les braises sur de longues distances, accélérant ainsi la progression du feu. Les températures élevées, la faible humidité relative et la sécheresse contribuent également à la propagation rapide des incendies.

### **5-2 Type de combustible**

La nature et la quantité de combustible disponible dans la forêt influencent la propagation des incendies. Les forêts riches en matières combustibles telles que les herbes sèches, les feuilles mortes, les aiguilles de pin et le sous-bois dense favorisent une combustion plus rapide et intense.

La teneur en humidité du combustible, qu'il s'agisse de végétation ou de bois mort, influence la facilité avec laquelle il prend feu. Des combustibles plus secs brûlent plus rapidement et génèrent des flammes plus intenses.

### **5-3 La topographie**

La topographie du terrain peut accélérer ou ralentir la propagation du feu. Les incendies ont tendance à se propager plus rapidement en montée qu'en descente en raison de la préchauffe du combustible sur les pentes. Les vallées peuvent agir comme des canaux qui canalisent le feu.

### **5-4 Intensité du feu initial**

L'intensité du feu initial, souvent déterminée par la source d'allumage, influence la manière dont le feu se propage. Un feu plus intense génère des courants d'air ascendants plus puissants, aspirant de l'air frais vers la base du feu et accélérant ainsi la combustion.

Comprendre ces facteurs est essentiel pour développer des stratégies de prévention et de gestion des incendies de forêt. Les autorités et les équipes de lutte contre les incendies

utilisent ces informations pour élaborer des plans d'intervention et minimiser les risques lors de la saison des incendies de forêt.

## 6- Les conséquences des incendies de forêts

Les incendies de forêts peuvent avoir de nombreuses conséquences, tant sur l'environnement que sur les communautés humaines. Voici quelques-unes des conséquences les plus courantes :

- **Impact sur la biodiversité** : Les incendies de forêts peuvent détruire des habitats naturels, entraînant la perte de biodiversité. Cela peut affecter les espèces végétales et animales, en particulier celles qui dépendent de la forêt pour leur survie. [8]
- **Changement climatiques** : Les incendies de forêts peuvent contribuer au changement climatique en libérant d'importantes quantités de dioxyde de carbone dans l'atmosphère.[9]
- **Qualité de l'air** : Les incendies de forêts produisent des particules fines et des composés chimiques qui peuvent avoir des effets néfastes sur la qualité de l'air, pouvant entraîner des problèmes respiratoires chez les populations humaines. [10]
- **Impacts socio-économiques** : Les incendies de forêts peuvent avoir des conséquences économiques importantes, notamment en détruisant des infrastructures, des terres agricoles et en affectant les secteurs du tourisme.[11]
- **Dégradation de sol** : Les incendies peuvent altérer la structure du sol, réduire sa fertilité et augmenter le risque d'érosion.[12]

Donc, les incendies de forêts représentent une menace significative pour l'environnement et les communautés humaines, engendrant un ensemble complexe de conséquences néfastes. Tous ces éléments soulignent l'importance cruciale de comprendre et de gérer les incendies de forêts de manière proactive. Des efforts concertés au niveau mondial sont nécessaires pour prévenir, atténuer et répondre à ces catastrophes, en intégrant des approches durables et des pratiques de gestion des incendies qui prennent en compte la protection de la biodiversité, la préservation des écosystèmes et la sécurité des communautés humaines.

## 7- Comment se prévenir contre les incendies de forêt

La prévention des incendies de forêts implique une combinaison de mesures à la fois préventives et éducatives. Voici quelques stratégies clés pour prévenir les incendies de forêts :

- **Surveillance et détection précoce** : Mettre en place un système de surveillance et de détection précoce des incendies, y compris l'utilisation de tours de guet, de capteurs et de technologies satellites pour identifier rapidement les départs de feu.
- **Entretien des forêts** : Pratiquer des activités d'entretien des forêts, telles que l'élagage des arbres morts et la suppression des broussailles, pour réduire la quantité de combustible disponible.
- **Gestion des combustibles** : Mettre en œuvre des stratégies de gestion des combustibles, telles que la création de coupe-feux et la réduction des matériaux inflammables à proximité des zones habitées.
- **Education du public** : Sensibiliser la population aux risques d'incendie de forêt, promouvoir des comportements responsables (comme l'extinction correcte des feux de camp) et informer sur les règles en vigueur en matière de prévention des incendies.
- **Législation et réglementation** : Élaborer et appliquer des lois et des règlements stricts en matière de prévention des incendies, y compris des restrictions pendant les périodes de sécheresse ou de conditions météorologiques extrêmes.
- **Formation des équipes de lutte contre les incendies** : Former et équiper les équipes de lutte contre les incendies pour qu'elles puissent intervenir rapidement et efficacement en cas d'incendie.
- **Planification urbaine** : Intégrer des zones tampons et des espaces ouverts dans la planification urbaine pour réduire le risque d'incendies de forêts atteignant les zones habitées.
- **Collaboration internationale** : Coopérer au niveau international pour partager les meilleures pratiques, les technologies et les ressources en matière de prévention et de gestion des incendies de forêts.

La combinaison de ces mesures contribue à créer un environnement plus résilient face aux incendies de forêts et à réduire les risques pour les écosystèmes et les communautés environnantes.

### 8- Exemples des incendies de forêts

Selon les données du site web "Global Forest Watch", qui compile des données sur les incendies de forêt dans le monde, 82 millions d'hectares de surface forestière ont brûlé dans le

monde au cours des dix dernières années, tandis que la plus grande perte de surface forestière a été enregistrée en 2016 avec la destruction de 9,63 millions d'hectares.

Alors que les plus grands incendies de ces dix dernières années ont eu lieu dans des pays tels que l'Australie, la Russie, le Canada et les États-Unis, la Grèce, l'Algérie et en Italie.[13].

Voici les incendies les plus célèbres qui ont frappé les forêts du monde au cours des dernières années.[14]

<b>Année</b>	<b>Pays</b>	<b>Superficie brûlé</b>	<b>victimes</b>
Mars 2022	Corée du Sud	6000 ha de forêt, et plus de 195 maisons	
Sept 2021	Algérie	Plus de 30.000 ha sur 30 Wilayas	Mort de 90 personnes
Juillet 2021	Italie	Plus de 800 incendies, La déclaration de l'état d'urgence dans le pays	
Août 2021	Turquie	Plus de 100.000 ha sur 46 gouvernorats	
Août 2021	Grèce	Plus 25000 ha et plus de 500 incendies	Mort 2 personnes
Août 2021	Russie	173000 km <sup>2</sup>	
Janv. 2020	Australie	10.000.000 ha et plus de 200 incendies. 200 maisons détruites	Mort de 28 personnes
Sept. 2020	USA	400.000 ha	Mort de 24 personnes
Oct. 2020	Syrie	30 incendies avec centaines d'hectares de forêts et arbre fruité	

*Table 1 : les incendies les plus grands au cours des dernières années*

## Conclusion

En conclusion, les incendies de forêts ne sont pas simplement des événements naturels, mais des phénomènes complexes aux conséquences multiples. La compréhension des types d'incendies, des facteurs de propagation et des conséquences associées est essentielle pour développer des stratégies de prévention et de gestion efficaces. Les impacts dévastateurs sur la biodiversité, le climat, la qualité de l'air, l'économie et les sols soulignent l'urgence de mesures proactives. La collaboration internationale, la sensibilisation publique, la réglementation stricte et la gestion appropriée des combustibles sont autant de pièces du puzzle pour atténuer les risques. En adoptant ces approches, nous pouvons non seulement prévenir les incendies de forêts, mais aussi préserver nos précieux écosystèmes et protéger les communautés humaines des impacts dévastateurs de ces catastrophes.

# **Chapitre 2**

## **Le concept d'intelligence artificielle**



### Introduction

Le domaine de l'intelligence artificielle (IA) représente une convergence fascinante entre l'informatique et les mathématiques, visant à créer des machines capables d'imiter l'intelligence humaine. Ce chapitre explore les fondements de l'IA, détaillant ses définitions, ses catégories, et son évolution historique. Nous aborderons également l'apprentissage machine, une sous-discipline essentielle de l'IA, en examinant ses types, son processus de fonctionnement, et les principaux algorithmes qui le sous-tendent.

### 1- C'est quoi l'intelligence artificielle ?

L'intelligence artificielle (IA) représente un domaine interdisciplinaire associant l'informatique et les mathématiques, englobant diverses techniques algorithmiques et théories visant à créer des machines capables d'imiter l'intelligence humaine. Son objectif principal est de reproduire l'intelligence afin de résoudre des problèmes complexes. Cette quête implique la modélisation de l'intelligence humaine en tant que phénomène, similaire à l'approche utilisée dans d'autres disciplines scientifiques telles que la physique, la chimie ou la biologie. L'IA, en constante expansion, trouve des fondements théoriques et des applications dans divers domaines tels que la théorie des probabilités, les neurosciences, la robotique, la théorie des jeux, la santé et les transports. L'intelligence artificielle est généralement catégorisée en deux types :

- **IA Forte** : Elle vise à développer des machines dotées de capacités cognitives approchant celles de l'humain. Ces machines devraient être capables de résoudre de manière autonome des problèmes qui leur sont inconnus. Bien que cela soit un domaine de recherche actif, l'IA forte n'est pas encore une réalité concrète.
- **IA Faible** : Son objectif est de créer des machines capables de résoudre des problèmes spécifiques. Ces machines disposent de données et bénéficient de l'aide humaine, mais elles doivent simuler le raisonnement humain sans forcément comprendre les phénomènes sous-jacents.

### 2- Définition de l'IA

Le terme « intelligence artificielle », créé par John McCarthy, est souvent abrégé par le sigle « IA » (ou « AI » en anglais, pour *artificial intelligence*) lors de la Conférence de Dartmouth en 1956. McCarthy définit l'IA ainsi : « C'est la science et l'ingénierie de la

fabrication de machines intelligentes, en particulier de programmes informatiques intelligents. Elle est liée à la tâche similaire qui consiste à utiliser des ordinateurs pour comprendre l'intelligence humaine, mais l'IA ne doit pas se limiter aux méthodes qui sont biologiquement observables. » [15]

Voici une définition plus formelle d'intelligence artificielle : "L'intelligence artificielle est une discipline de l'informatique qui vise à créer des systèmes capables de réaliser des tâches qui nécessitent généralement l'intelligence humaine. Ces tâches incluent la résolution de problèmes complexes, la reconnaissance de motifs, l'apprentissage automatique, la compréhension du langage naturel, la vision par ordinateur, la prise de décision, et d'autres capacités cognitives." [16]

### 3- Historiques et chronologique de l'IA

L'histoire de l'intelligence artificielle (IA) remonte à plusieurs décennies et est marquée par des développements significatifs. Voici une chronologie simplifiée des moments clés de l'histoire de l'IA : [17]

#### ➤ **Années 1940-1950 : Les débuts**

- 1943 : Warren McCulloch et Walter Pitts proposent un modèle de neurones formels.
- 1950 : Alan Turing introduit le "Test de Turing" comme un moyen de mesurer l'intelligence d'une machine.

#### ➤ **Années 1950-1960 : La naissance de l'IA**

- 1956 : La conférence de Dartmouth, considérée comme le début officiel de l'intelligence artificielle.
- 1956-1974 : Période de l'«optimisme initial», avec des travaux sur les réseaux neuronaux, le langage naturel et les jeux.

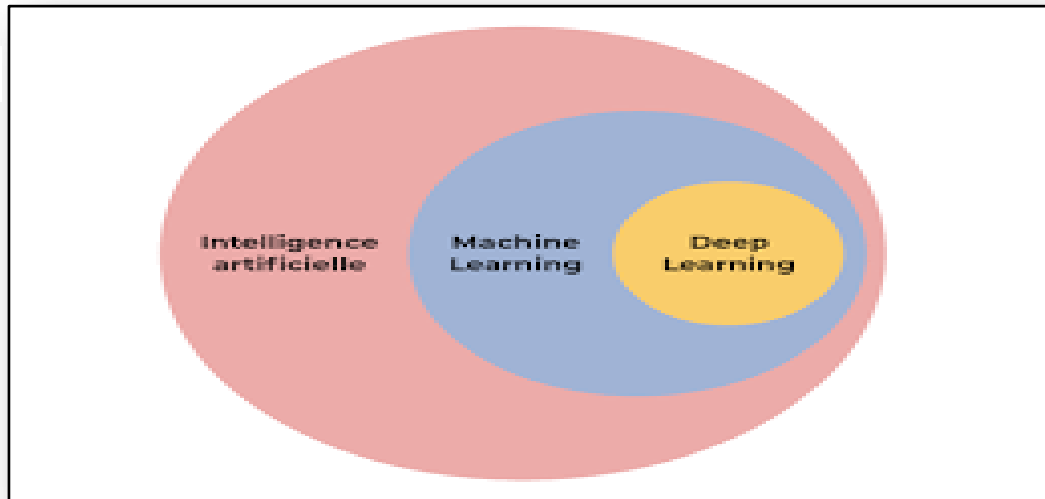
#### ➤ **Années 1960-1970 : La première vague de l'IA**

- 1965 : ELIZA, un programme de traitement du langage naturel, est développé par Joseph Weizenbaum.
- 1969 : Marvin Minsky et Seymour Papert publient "Perceptrons", critiquant certaines limitations des réseaux neuronaux.

- **Années 1970-1980 : L'IA connaît des revers**
  - Les capacités des ordinateurs et donc les programmes d'IA étaient assez limités par rapport à ceux d'aujourd'hui. Même les meilleurs programmes ne pouvaient traiter que des versions simples des problèmes qu'ils étaient censés résoudre.
- **Années 1980-2000 : Résurgence de l'IA**
  - 1980s : la croissance simultanée de la robotique a influencé de manière significative l'IA, introduisant un nouveau paradigme plaidant pour l'importance de donner un corps aux ordinateurs pour manifester une intelligence réelle.
  - 1997 : IBM's Deep Blue bat le champion du monde d'échecs Garry Kasparov.
  - Fin des années 1990 : L'apprentissage automatique et les algorithmes de support vectoriel gagnent en popularité.
- **Années 2000-2010 : L'émergence du Big Data et de l'apprentissage profond**
  - 2009 : Début de l'ère de l'apprentissage profond avec des réseaux de neurones profonds.
  - 2011 : IBM's Watson remporte le jeu télévisé Jeopardy!.
- **Années 2010-2020 : L'IA devient omniprésente**
  - 2014 : Google DeepMind développe un réseau neuronal qui bat les champions humains au jeu de société Go.
  - 2018 : Développement rapide de l'IA dans des domaines tels que la vision par ordinateur, la traduction automatique et les voitures autonomes.
- **Années 2020 et au-delà : Évolutions continues**
  - L'IA continue de se développer avec des avancées majeures dans des domaines tels que l'apprentissage automatique, le traitement du langage naturel et la robotique.

L'histoire de l'IA est caractérisée par des périodes d'enthousiasme et de défis, mais elle continue d'évoluer rapidement, façonnant de nombreux aspects de notre vie quotidienne et de divers secteurs.

Le domaine de l'intelligence artificielle se divise en plusieurs sous-domaines entrelacés, tels que présentés dans *la figure 3*. L'apprentissage automatique (*Machine Learning*), l'un de ces sous-domaines, consiste à acquérir des règles implicites par l'expérience ou à partir de bases de données pour résoudre des problèmes donnés.



*Figure 3 : les sous domaine de l'IA*

#### 4- L'apprentissage automatique

En 1959, Arthur Samuel, pionnier du jeu sur ordinateur, définissait le machine learning comme « un domaine d'études qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmés ».[18]

L'Apprentissage Machine (Machine Learning) est une branche de l'intelligence artificielle qui permet aux ordinateurs d'apprendre à partir de données sans être explicitement programmés. Cela se fait en utilisant des algorithmes qui permettent aux systèmes informatiques d'analyser des données, d'identifier des motifs et de prendre des décisions avec un minimum d'intervention humaine.

#### 5- Le processus de fonctionnement du ML

Le processus du Machine Learning implique plusieurs étapes clés qui permettent à un modèle d'apprentissage automatique d'acquérir des connaissances à partir de données. Voici une description générale du processus de fonctionnement du Machine Learning :

- **Collecte des données** : Le processus commence par la collecte de données pertinentes pour le problème à résoudre. Ces données peuvent être structurées (par exemple, des tableaux de base de données) ou non structurées (par exemple, des images, des textes).
- **Prétraitement des données** : Les données collectées peuvent nécessiter un nettoyage et une préparation. Cela peut inclure la gestion des valeurs manquantes, la normalisation des données, la conversion de formats, etc.

- **Choix du modèle** : En fonction de la nature du problème (classification, régression, clustering, etc.), un modèle d'apprentissage est choisi. Les modèles peuvent inclure des algorithmes tels que les réseaux de neurones, les machines à vecteurs de support, les arbres de décision, etc.
- **Division des données** : Les données sont généralement divisées en ensembles d'entraînement (pour former le modèle) et en ensembles de test (pour évaluer sa performance).
- **Entraînement du modèle** : Le modèle est alimenté avec les données d'entraînement, et il ajuste ses paramètres pour minimiser l'écart entre les prédictions et les vraies valeurs. Cela se fait en utilisant des algorithmes d'optimisation qui ajustent itérativement les poids du modèle.
- **Validation et ajustement** : Le modèle est évalué sur l'ensemble de validation pour s'assurer qu'il généralise bien aux données qu'il n'a pas vues pendant l'entraînement. Si nécessaire, les hyper paramètres du modèle peuvent être ajustés pour améliorer les performances.
- **Évaluation du modèle** : Une fois le modèle entraîné, il est évalué sur l'ensemble de test pour estimer sa performance réelle sur des données inconnues. Cela permet de s'assurer que le modèle peut généraliser à de nouvelles situations.
- **Prédiction** : Une fois validé et évalué, le modèle peut être utilisé pour faire des prédictions sur de nouvelles données, fournissant des résultats ou des classifications basées sur l'apprentissage effectué pendant la phase d'entraînement.

### 6- Les types d'Apprentissage Machine

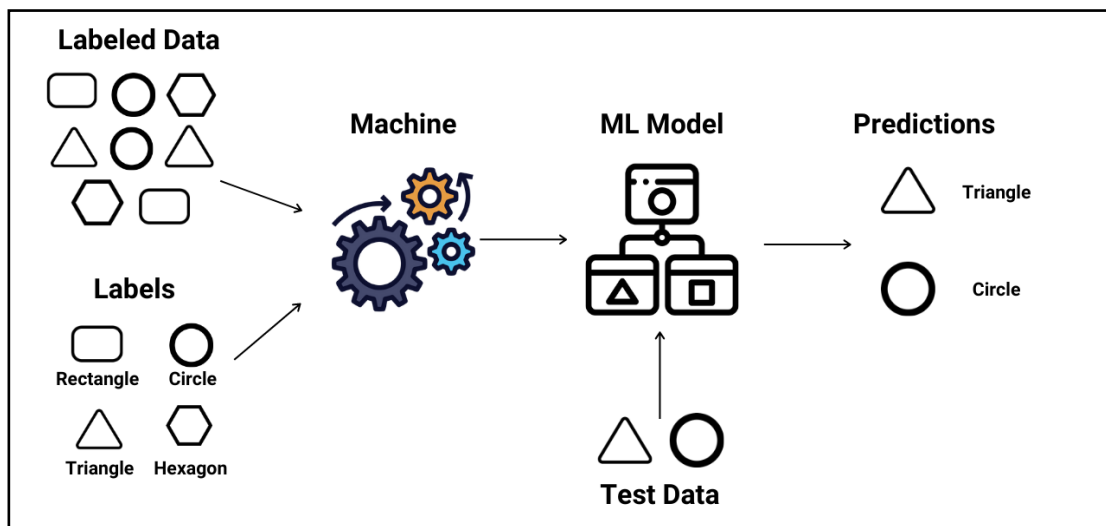
Il est important de noter que le Machine Learning peut être supervisé, non supervisé, semi-supervisé ou par renforcement, en fonction de la nature des données d'entraînement et du type de problème à résoudre. Chaque type d'apprentissage a ses propres méthodes et algorithmes spécifiques.[19]

#### 6-1 Apprentissage Supervisé

Dans l'apprentissage supervisé, le modèle est entraîné sur un ensemble de données étiquetées où les entrées et les sorties sont explicitement associées. Le modèle apprend à faire des prédictions en comparant ses résultats avec les étiquettes de données.

L'apprentissage automatique supervisé permet d'effectuer des prévisions sur des données non encore disponibles ou futures, ce qu'on appelle la "modélisation prédictive". L'algorithme s'efforce de créer une fonction capable de prédire de manière précise la sortie en se basant sur les variables d'entrée. Par exemple, il peut anticiper la valeur d'un bien immobilier (sortie) en fonction de données telles que le nombre de pièces, l'année de construction, la superficie du terrain, et l'emplacement. Le Machine Learning supervisé peut se subdiviser en deux types :

- **Classification** : la variable de sortie est une catégorie.
- **Régression** : la variable de sortie est une valeur spécifique.



*Figure4 : Apprentissage supervisé*

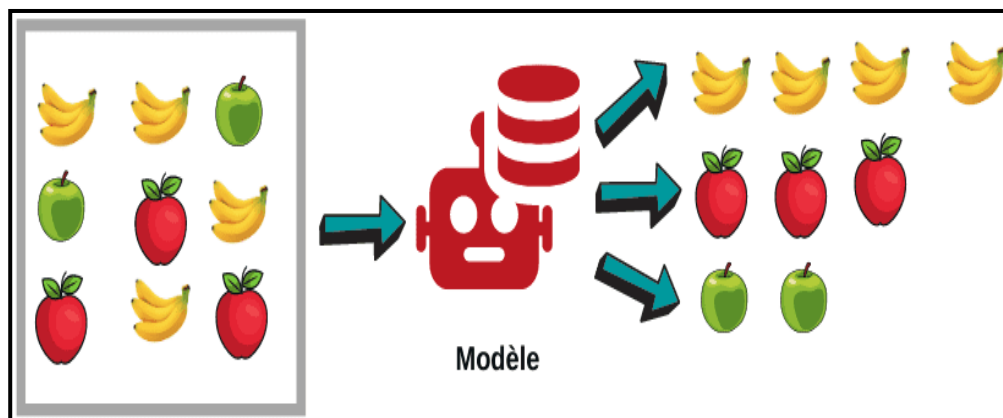
Un exemple classique d'apprentissage supervisé est celui de la classification. Dans ce scénario, l'algorithme est formé sur un ensemble de données étiquetées où les exemples sont associés à des catégories spécifiques. L'objectif est d'apprendre à classer de nouveaux exemples non étiquetés en fonction des modèles identifiés pendant l'entraînement.

Prenons l'exemple de la classification des e-mails en tant que "*spam*" ou "*non-spam*". Dans un ensemble de données d'apprentissage, chaque e-mail est étiqueté comme étant soit "*spam*" soit "*non-spam*". L'algorithme est ensuite formé sur ces données en apprenant les caractéristiques associées à chaque catégorie. Une fois entraîné, l'algorithme peut être utilisé pour classer automatiquement de nouveaux e-mails comme "*spam*" ou "*non-spam*" en fonction des schémas qu'il a appris pendant l'entraînement.

## 6-2 Apprentissage Non Supervisé

À l'inverse, l'apprentissage non supervisé implique l'utilisation d'ensembles de données non étiquetées. L'algorithme détermine lui-même la structure de l'entrée, sans l'application préalable d'étiquettes. Cette approche peut être une finalité en elle-même, permettant la découverte de structures cachées dans les données, ou bien un moyen d'atteindre un objectif spécifique. On la désigne également sous le terme d'apprentissage des caractéristiques (*feature learning*). Il existe deux types de Machine Learning non supervisé :

- **Clustering** : l'objectif consiste à trouver des regroupements dans les données.
- **Association** : l'objectif consiste à identifier les règles qui permettront de définir de grands groupes de données.



*Figure 5: Apprentissage non supervisé*

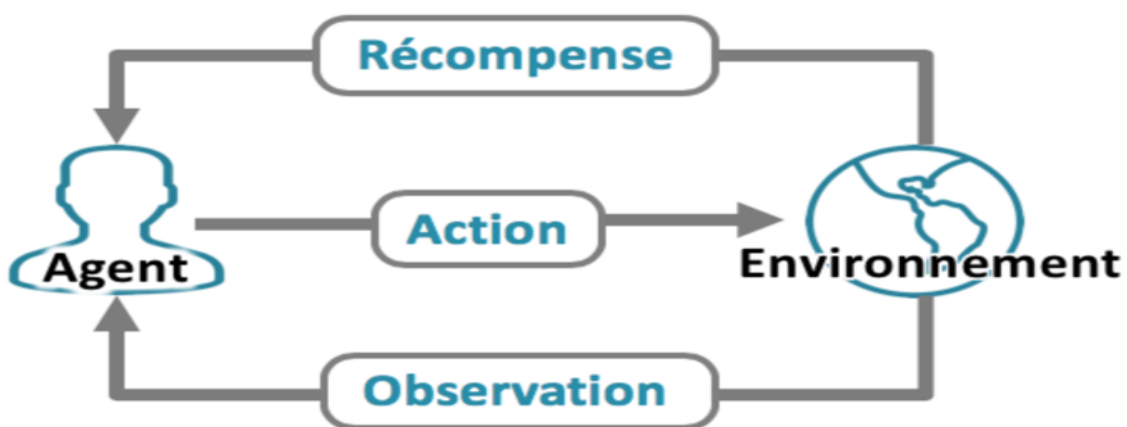
Un exemple courant d'apprentissage non supervisé est le regroupement (clustering). Dans ce scénario, l'algorithme explore des données sans avoir de pré-étiquettes associées. Au lieu de cela, il cherche à identifier des similarités ou des structures intrinsèques dans les données pour regrouper des éléments similaires. Par exemple, imaginez que vous ayez un ensemble de données comprenant des caractéristiques telles que la taille, le poids, et la couleur de différents fruits. Sans informations préalables sur les types de fruits, un algorithme d'apprentissage non supervisé pourrait regrouper les fruits en fonction de leurs caractéristiques similaires. Ainsi, il pourrait identifier spontanément des groupes qui correspondent aux pommes, aux oranges, et aux bananes, même sans avoir été préalablement informé des catégories spécifiques de fruits.

### 6-3 L'apprentissage semi-supervisé

L'apprentissage semi-supervisé est une approche dans le domaine de l'apprentissage automatique où un modèle est entraîné à partir d'un ensemble de données qui contient à la fois des exemples étiquetés (données avec des réponses connues) et des exemples non étiquetés (données sans réponses). Contrairement à l'apprentissage supervisé où le modèle est entraîné uniquement sur des données étiquetées, l'apprentissage semi-supervisé vise à exploiter les informations contenues dans les données non étiquetées pour améliorer les performances du modèle. Cette approche est particulièrement utile dans des situations où l'acquisition d'un grand ensemble de données étiquetées peut être coûteuse ou difficile, mais où des données non étiquetées sont disponibles en abondance. En tirant parti de ces données non étiquetées, l'apprentissage semi-supervisé vise à améliorer la généralisation du modèle et sa capacité à traiter de nouvelles données. Cela peut être particulièrement bénéfique lorsque le coût de l'annotation des données est prohibitif ou lorsque les données étiquetées sont rares.

### 6-4. Apprentissage par Renforcement

L'apprentissage par renforcement implique un agent qui prend des actions dans un environnement pour atteindre un objectif particulier. L'agent apprend en recevant des récompenses ou des sanctions pour ses actions.



*Figure 6 : Apprentissage par renforcement*

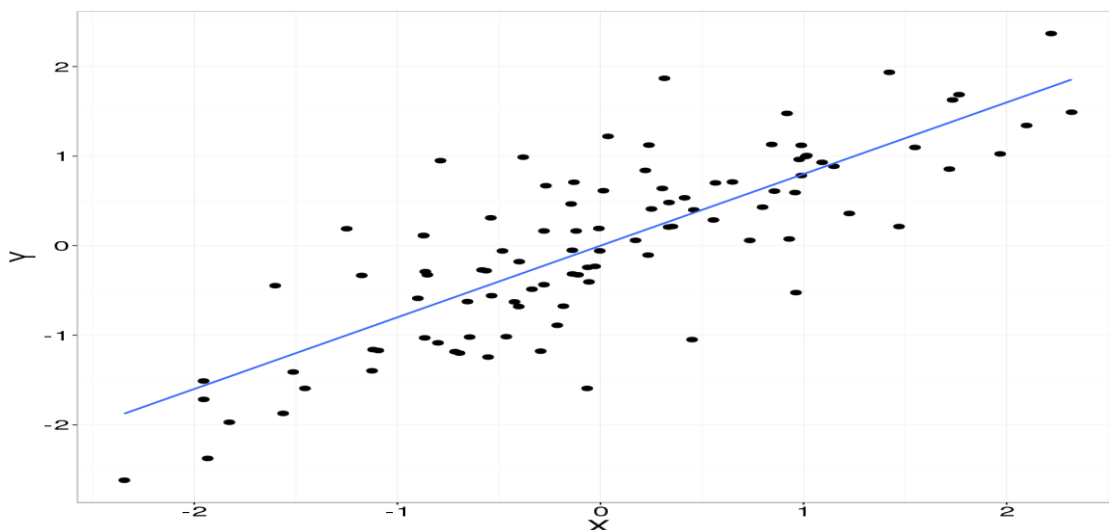
Imaginons un exemple où un drone doit apprendre à naviguer dans un environnement complexe en utilisant l'apprentissage par renforcement. L'objectif est de permettre au drone d'optimiser ses actions pour maximiser une récompense définie en fonction de sa performance. Voici comment cela pourrait fonctionner :

- **Agent** : Le drone, qui prend des décisions (actions) basées sur son état actuel pour maximiser la récompense totale au fil du temps
- **Environnement** : Un environnement simulé ou réel où le drone peut se déplacer, éviter les obstacles, atteindre des destinations spécifiques, etc.
- **Etat** : Les informations sensorielles du drone, telles que les données provenant des capteurs (caméras, lidars, gyroscopes), sa position, la présence d'obstacles, etc.
- **Action** : Les commandes que le drone peut exécuter, comme accélérer, tourner, monter, descendre, etc.
- **Récompense** : Une fonction qui évalue la performance du drone. Par exemple, le drone pourrait recevoir une récompense positive lorsqu'il atteint une destination et une récompense négative s'il percute un obstacle.
- **Apprentissage** : Utilisation d'un algorithme d'apprentissage par renforcement, comme le Q-Learning ou les méthodes basées sur des politiques (Policy Gradient), pour que le drone apprenne à ajuster ses actions en fonction de ses expériences pour maximiser la récompense cumulative.

### 7- Principaux algorithmes de Machine Learning

#### 7-1 Régression Linéaire

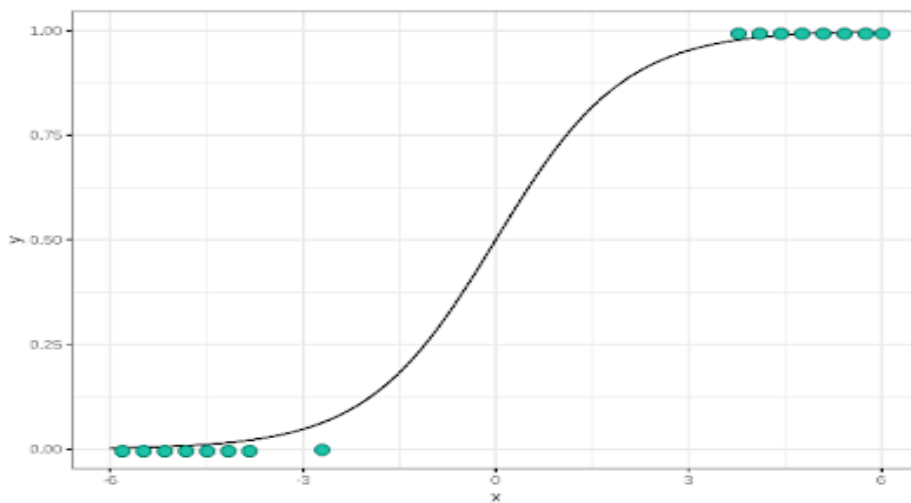
Les **algorithmes de régression linéaire** affichent ou prédisent la relation entre deux variables ou facteurs en ajustant une ligne droite continue aux données. La ligne est souvent calculée à l'aide de la fonction de coût d'erreur quadratique. La régression linéaire est l'un des types d'analyse de régression les plus populaire.



*Figure 7 : Algorithme de régression linéaire*

### 7-2 Algorithmes de régression logistique

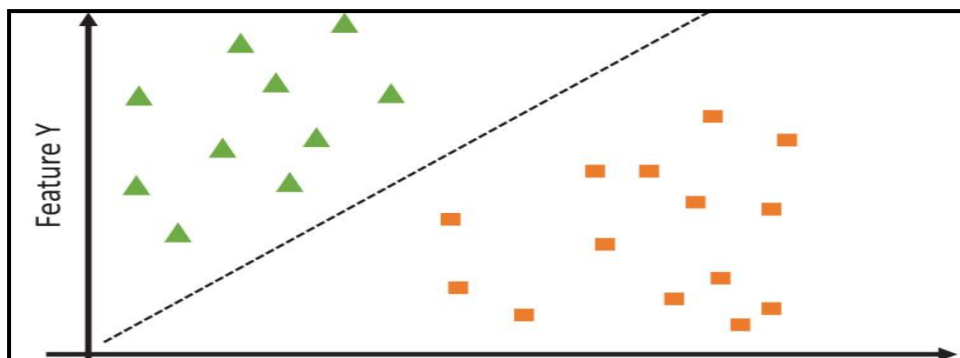
Les **algorithmes de régression logistique** ajustent une courbe en forme de S continue aux données. La régression logistique est un autre type populaire d'analyse de régression. Un modèle de régression logistique permet aussi de prédire la probabilité qu'un événement arrive (valeur de 1) ou non (valeur de 0) à partir de l'optimisation des coefficients de régression. Ce résultat varie toujours entre 0 et 1. Lorsque la valeur prédite est supérieure à un seuil, l'événement est susceptible de se produire, alors que lorsque cette valeur est inférieure au même seuil



*Figure 8 : Algorithme de régression logistique*

### 7-3 Support Vector Machines (SVM)

Les machines à vecteurs de support ou SVM (Séparateurs à vastes marges) sont des algorithmes qui séparent les données en classes. Pendant l'entraînement, un SVM trouve une ligne qui sépare les données d'un jeu en classes spécifiques et maximise les marges (les distances entre les frontières de séparation et les échantillons les plus proches) de chaque classe. Après avoir appris les lignes de classification, le modèle peut ensuite les appliquer aux nouvelles données.



*Figure 9: Algorithme Support Vector Machines*

### 7-4 Arbres de Décision

Un algorithme d'arbre de décision représente graphiquement les données en branches pour montrer les résultats possibles de diverses actions. Il classe et prédit les variables de réponse en fonction des décisions passées. Les résultats des arbres de décision sont faciles à expliquer. Cependant, les arbres de décisions deviennent difficiles à lire quand ils sont associés à de gros volumes de données et à des variables complexes.

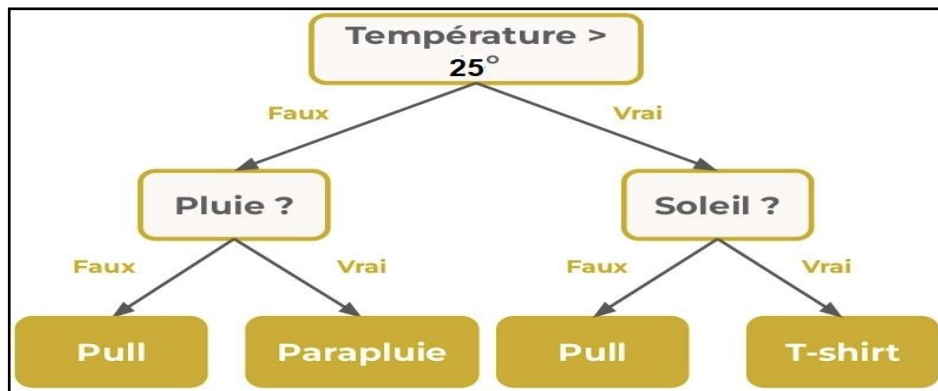


Figure 10: Algorithme arbre de décision

### 7-5 Algorithme K-Means

**K-Means** est un algorithme non supervisé de clustering. Il permet de regrouper en clusters distincts les observations du data set. Ainsi les données similaires se retrouveront dans un même cluster. Par ailleurs, une observation ne peut se retrouver que dans un cluster à la fois (exclusivité d'appartenance). Une même observation, ne pourra donc, appartenir à deux clusters différents. Pour pouvoir regrouper un jeu de données en cluster distincts, l'algorithme K-Means a besoin d'un moyen de **comparer le degré de similarité** entre les différentes observations. Ainsi, deux données qui se ressemblent, auront une **distance de dissimilarité** réduite, alors que deux objets différents auront une distance de séparation plus grande.

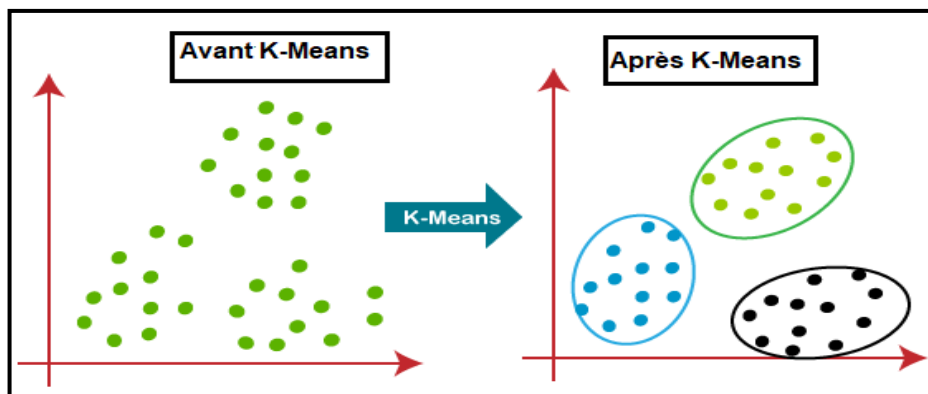
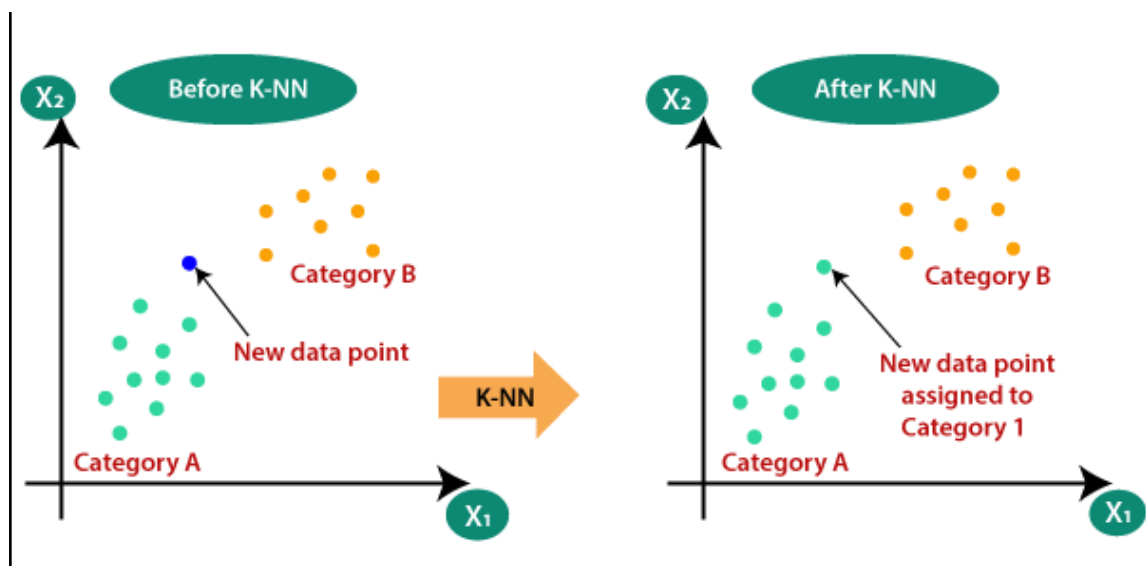


Figure 11: Algorithme K-Means

### 7-6 Algorithme K-Nearest Neighbors(KNN)

L'algorithme des K plus proches voisins est un algorithme de Machine Learning qui appartient à la classe des algorithmes d'apprentissage supervisé simple et facile à mettre en œuvre qui peut être utilisé pour résoudre les problèmes de classification et de régression. L'intuition derrière l'algorithme des KNN est l'une des plus simples de tous les algorithmes de Machine Learning supervisé :

- Sélectionnez le nombre K de voisins ;
- Calculez la distance du point non classifié aux autres points ;
- Prenez les K voisins les plus proches selon la distance calculée ;
- Parmi ces K voisins, comptez le nombre de points appartenant à chaque catégorie ;
- Attribuez le nouveau point à la catégorie la plus présente parmi ces K voisins ;
- Notre modèle est prêt :



*Figure 12: Algorithme K-Nearest Neighbors*

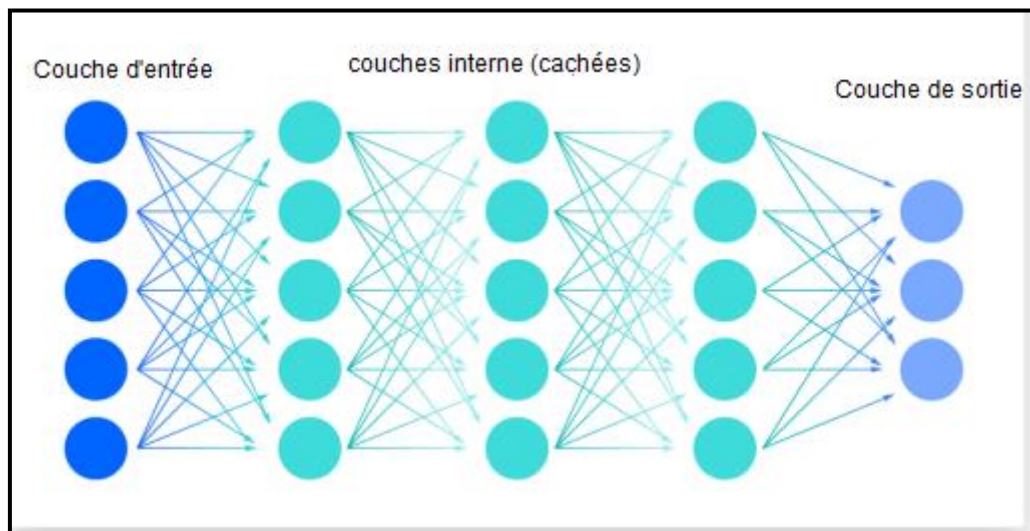
### 7-8 Algorithme Q-Learning

Le Q-learning est un algorithme d'apprentissage par renforcement qui cherche à trouver la meilleure action à entreprendre compte tenu de l'état actuel. Il est considéré comme hors politique parce que la fonction de Q-learning apprend des actions qui sont en dehors de la politique actuelle, comme prendre des actions aléatoires, et donc une politique n'est pas nécessaire. Plus précisément, le Q-learning cherche à apprendre une politique qui maximise la récompense totale.

Le « Q » de Q-learning est synonyme de qualité. Dans ce cas, la qualité représente l'utilité d'une action donnée pour obtenir une récompense future.

### 8- Apprentissage profond (Deep Learning)

Le deep learning, ou apprentissage profond en français, est une sous-discipline de l'intelligence artificielle (IA) qui se concentre sur l'utilisation de réseaux de neurones artificiels profonds pour effectuer des tâches complexes d'apprentissage et de reconnaissance de modèles [20]. Ces réseaux de neurones, structurés en plusieurs couches, permettent au système d'apprendre des représentations hiérarchiques de données, en identifiant des caractéristiques de plus en plus abstraites au fur et à mesure qu'il progresse dans les couches du réseau. Le deep learning a connu un grand succès dans des domaines tels que la vision par ordinateur, le traitement du langage naturel, la reconnaissance vocale et d'autres applications nécessitant une compréhension sophistiquée des données.



*Figure 13 : Réseau de neurones artificiels*

En voici les principaux algorithmes de deeplearning :

#### 8-1 Réseaux de Neurones Convolutifs (CNN)

Les CNN se composent de nombreuses couches conçues pour traiter et extraire des caractéristiques des données. En particulier, ces réseaux neuronaux convolutifs sont employés pour analyser et détecter des objets. Le CNN est excellent pour l'extraction de caractéristiques et la classification. Il est devenu prédominant dans la reconnaissance d'images, la classification et la reconnaissance vidéo.[21]

### 8-2 Réseaux de Neurons Récurrents (RNN)

Les réseaux de neurones récurrents (RNN) sont une architecture spécifique de réseaux de neurones artificiels conçue pour travailler avec des données séquentielles. Contrairement aux réseaux de neurones classiques, les RNN ont des connexions récurrentes qui leur permettent de prendre en compte les informations antérieures dans le traitement des données séquentielles. Cette capacité à mémoriser et à utiliser des informations passées est particulièrement utile dans des tâches telles que le traitement du langage naturel, la reconnaissance de séquences temporelles, et d'autres applications où la compréhension du contexte est essentielle.[21]

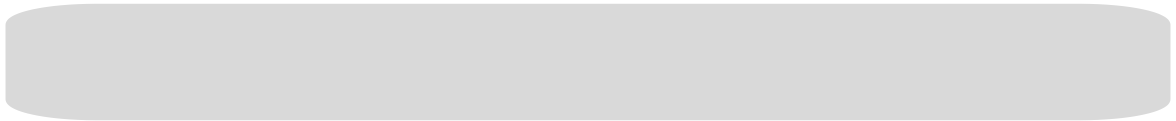
### Conclusion :

En résumé, ce chapitre offre une plongée approfondie dans le vaste monde de l'intelligence artificielle et de l'apprentissage machine. Depuis les débuts de l'IA jusqu'à son omniprésence contemporaine, en passant par les divers types d'apprentissage et ses algorithmes, chacune de ces étapes contribue à façonner un paysage technologique en constante évolution. En comprenant les bases théoriques et les applications pratiques de l'IA.

Dans le prochain chapitre on se concentrera sur une exposition approfondie de notre approche, qui joue un rôle crucial dans la prévision des incendies de forêt en utilisant l'apprentissage automatique supervisé.

# **Chapitre 3**

## **Les datasets et les algorithmes choisis**



### Introduction

Les incendies de forêt représentent une menace sérieuse pour les écosystèmes forestiers, la biodiversité, ainsi que pour les populations et les infrastructures environnantes. Dans ce contexte, le développement de systèmes efficaces de prédiction et de détection des incendies de forêt revêt une importance cruciale pour la gestion et la prévention de ces catastrophes naturelles.

Ce chapitre se focalise sur la présentation des datasets utilisés ainsi que sur les algorithmes de machine learning sélectionnés pour élaborer un système de prédiction et de détection des incendies de forêt. En effet, la qualité des données et le choix des méthodes d'analyse sont des éléments déterminants pour la fiabilité et l'efficacité du système développé.

L'objectif principal de ce chapitre est de fournir une vue d'ensemble détaillée des datasets utilisés pour notre étude, en mettant en lumière ses source, ses structure et les différentes étapes de prétraitement. De plus, nous présenterons les algorithmes de ML sélectionnés, justifiant ainsi nos choix et détaillant leur adaptation à la problématique spécifique de prédiction et de détection des incendies de forêt.

### 1- Description des datasets

Pour le développement d'un système de prédiction et de détection des incendies de forêt, nous disposons de deux ensembles de données distincts, chacun jouant un rôle crucial dans le processus. Le premier dataset est spécifiquement dédié à la prédiction des incendies de forêt, fournissant des informations précieuses telles que les conditions météorologiques, la topographie, et les données historiques d'incendies. En parallèle, le deuxième dataset est axé sur la détection des incendies. L'intégration de ces deux ensembles de données complémentaires est essentielle pour fournir une solution holistique, à la fois prédictive et réactive, dans la lutte contre les incendies de forêt.

#### 1-1 Dataset pour la prédiction de magnitude des incendies (Dataset N°1)

Ce dataset présenté ici est un sous-échantillon d'une base de données contenant 1,88 million d'incendies survenus aux États-Unis de 1992 à 2015[22]. Les enregistrements des incendies de forêt ont été obtenus à partir des systèmes de déclaration des organisations fédérales, étatiques et locales chargées de la lutte contre les incendies de forêt. Une vérification de base des erreurs a été effectués et les enregistrements redondants ont été identifiés et supprimés. Le produit résultant, appelé *base de données d'occurrence d'incendie*

du programme d'analyse des incendies (FPA FOD), comprend 1,88 million d'enregistrements d'incendies de forêt géoréférencés, représentant un total de 140 millions d'acres brûlés au cours de la période de 24 ans. Ce dataset est combiné aussi avec des données météorologiques historiques à une latitude/longitude spécifique [23] et des données historiques sur la végétation. [24]

Les attributs du jeu de données sont clairement expliqués dans le tableau suivant. Ce tableau détaille chaque attribut, fournissant des descriptions précises et complètes qui facilitent la compréhension et l'utilisation du jeu de données.

N°	Attribut	Description
01	Cause	Cause de l'incendie
02	Latitude	Latitude de l'endroit de feu
03	Longitude	Longitude de l'endroit feu
04	Date	Date de l'incendie
05	Vegitation	Végétation dominante dans les zones
06	Temp_pre_30	La moyenne de température en degrés C sur le lieu de l'incendie jusqu'à 30 jours avant
07	Temp_pre_15	La moyenne de température en degrés C sur le lieu de l'incendie jusqu'à 15 jours avant
08	Temp_pre_7	La moyenne de température en degrés C sur le lieu de l'incendie jusqu'à 07 jours avant
09	Temp_cont	La température en degrés C sur le lieu de l'incendie au jour de l'incendie
10	Wind_pre_30	vent en m/s sur le lieu de l'incendie jusqu'à 30 jours avant
11	Wind_pre_15	vent en m/s sur le lieu de l'incendie jusqu'à 15 jours avant
12	Wind_pre_7	vent en m/s sur le lieu de l'incendie jusqu'à 07 jours avant
13	Wind_cont	vent en m/s sur le lieu de l'incendie au jour de l'incendie
14	Hum_pre_30	humidité en % sur le lieu de l'incendie jusqu'à 30 jours avant
15	Hum_pre_15	humidité en % sur le lieu de l'incendie jusqu'à 15 jours avant
16	Hum_pre_7	humidité en % sur le lieu de l'incendie jusqu'à 07 jours avant
17	Hum_cont	humidité en % sur le lieu de l'incendie au jour de l'incendie
18	Prec_pre_30	précipitations en mm sur le lieu de l'incendie jusqu'à 30 jours avant

19	Prec_pre_15	précipitations en mm sur le lieu de l'incendie jusqu'à 15 jours avant
20	Prec_pre_7	précipitations en mm sur le lieu de l'incendie jusqu'à 07 jours avant
21	Prec_cont	précipitations en mm sur le lieu de l'incendie au jour de l'incendie
22	Fire_size	Taille du feu en Acre (1 Acre $\approx$ 0,40 Ha)
23	Fire_class	Classe du feu : A, B, C ou D

*Table 2 : Les attributs du dataset*

Le tableau suivant offre une explication détaillée des différentes causes des incendies, simplifiant ainsi leur compréhension et leur interprétation pour une analyse plus approfondie.

N°	Cause	Description
01	Arson	L'incendie criminel implique l'allumage intentionnel d'un feu dans le but de causer des dommages ou de nuire.
02	Campfire	Un feu de camp est un feu utilisé en plein air, soit pour la cuisson, soit pour fournir de la chaleur ou une ambiance.
03	Children	Un incendie déclenché par des enfants, souvent accidentellement, en raison de leur curiosité naturelle, de leur ignorance des dangers du feu.
04	Debris burning	Brûler des matériaux de déchets tels que des feuilles, des branches et d'autres matières organiques généralement collectées lors de l'entretien des jardins ou des cours.
05	Equipment use	Feux déclenchés par des équipements mécaniques comme des tronçonneuses et les débroussailleuses.
06	Lightning	Un éclair est une décharge électrique de grande intensité qui se produit dans l'atmosphère, généralement pendant un orage.
07	Miscellaneous	Des causes diverses.
08	Powerline	Les lignes électriques à haute tension.
09	Railroad	Les chemins de fer.
10	Smoking	incendie qui a été déclenché par un objet enflammé, tel qu'une cigarette, une pipe ou un cigare, qui a été mal éteint.
11	Firewoks	L'utilisation de feux d'artifice peut déclencher des incendies de forêt, en particulier dans des conditions sèches et venteuses
12	Structures	Les incendies qui commencent dans des structures telles que des bâtiments et des usines. Les braises ou les flammes transportées par le vent peuvent enflammer la végétation alentour.
13	Undefined	Causes indéterminées.

*Table 3 : Causes de déclenchement d'un incendie*

Les incendies de ce jeu de données sont classés en quatre catégories selon la superficie brûlée. Cette classification permet de mieux comprendre et analyser l'ampleur de chaque incendie, facilitant ainsi les efforts de prédiction et de gestion des ressources. Le tableau suivant (Table 4) présente ces quatre classes

Classe	La superficie brûlée (en Acre)
A	Inférieur ou égal à 10
B	Entre 10 et 300
C	Entre 300 et 5000
D	Plus ou égal à 5000

*Table 4 : les classes des incendies*

### 1-2 Dataset pour la détection des incendies (Dataset N°2)

C'est un ensemble d'images organisé pour résoudre le problème de détection des incendies de forêt. Toutes les images de l'ensemble de données sont à 3 canaux avec une résolution de  $250 \times 250$ . Les images ont été récupérées en recherchant divers termes de recherche dans plusieurs moteurs de recherche. Ensuite, ces images sont étudiées minutieusement pour recadrer et supprimer les éléments inappropriés tels que les personnes, les machines d'extinction d'incendie, etc. afin de garantir que chaque image ne contient que la région d'incendie concernée. L'ensemble de données est conçu pour le problème binaire de détection d'incendie et d'absence d'incendie (*fire/nofire*) dans le paysage forestier. Il s'agit d'un ensemble de données équilibré composé de 1 900 images au total, dont 950 images appartiennent à chaque classe. L'ensemble de données est divisé en 80 :20 à des fins de formation et de test dans l'étude proposée.[25]

La figure 14 ci-dessous présente un extrait représentatif de l'ensemble d'images de notre dataset.



*Figure 14 : extrait de l'ensemble des images*

## 2-Choix des algorithmes

Dans le cadre de notre projet visant à améliorer la prédiction et la détection des incendies de forêt grâce à l'intelligence artificielle, nous prévoyons d'exploiter deux algorithmes de ML particulièrement adaptés à ces tâches. Premièrement, nous choisissons l'algorithme des K-Nearest Neighbors (K-NN) pour la prédiction de magnitude et classification des incendies. L'approche K-NN, reconnue pour sa simplicité et son efficacité dans les problèmes de classification et de régression, permettra d'analyser et de prédire les zones à risque en se basant sur les données historiques et environnementales. Ensuite, pour la détection des incendies en temps réel, nous nous tournons vers les réseaux de neurones convolutifs (CNN). Les CNN sont particulièrement efficaces pour traiter les données visuelles, ce qui les rend idéaux pour analyser les images satellitaires et détecter les débuts d'incendie. Cette combinaison d'approches offre un équilibre entre prédiction basée sur des données historiques et détection immédiate, optimisant ainsi nos chances de combattre efficacement les incendies de forêt.

### 2.1 Algorithme K-Nearest Neighbors KNN

L'algorithme des k plus proches voisins(K-Nearest Neighbors) appartient à la famille des algorithmes d'apprentissage automatique (ML).L'idée d'apprentissage automatique ne date pas d'hier, puisque le terme de ML a été utilisé pour la première fois par l'informaticien américain Arthur Samuel en 1959.[26]

L'algorithme des k plus proches voisins est une méthode d'apprentissage supervisée, qui utilise la proximité pour effectuer des classifications ou des prédictions sur le regroupement d'un point de données individuel. Bien qu'il puisse être utilisé pour des problèmes de régression ou de classification, il est généralement utilisé comme algorithme de classification, en partant de l'hypothèse que des points similaires peuvent être trouvés les uns à côté des autres.

Son fonctionnement peut être assimilé à l'analogie suivante :

*« Dis-moi qui sont tes voisins, je te dirais qui tu es .. ! »*

#### 2.1.1 Principe

Pour effectuer une prédiction, le KNN se base sur le jeu de données en entier. Pour une observation qui ne fait pas partie du jeu de données et qu'on souhaite prédire, l'algorithme va chercher les K instances du jeu de données les plus proches de notre observation. Ensuite pour ces K voisins, l'algorithme se basera sur leurs variables de sortie (Y) pour calculer la valeur de la variable de sortie (Y) de l'observation qu'on souhaite prédire.[27]

- Si KNN est utilisé pour la régression, c'est la moyenne (ou la médiane) des variables (Y) des K plus proches observations qui servira pour la prédiction.
- Si KNN est utilisé pour la classification ; c'est le mode des variables (Y) des K plus proches observations qui servira pour la prédiction.

Voici l'algorithme :

#### Début Algorithme

##### **Données en entrée :**

- un ensemble de données D .
- une fonction de définition distance  $d$ .
- Un nombre entier K

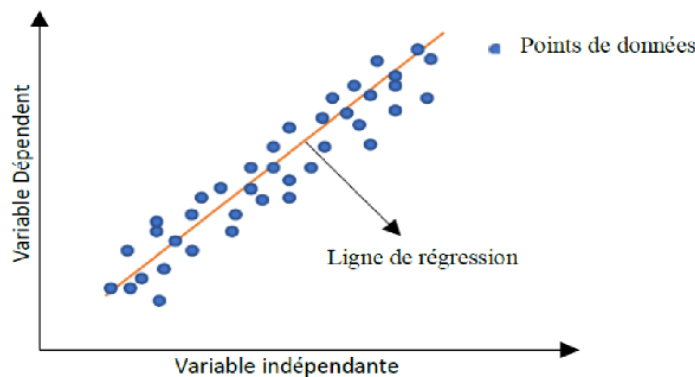
**Pour une nouvelle observation X dont on veut prédire sa variable de sortie y Faire :**

1. Calculer toutes les distances de cette observation  $X$  avec les autres observations du jeu de données  $D$
2. Retenir les  $K$  observations du jeu de données  $D$  les proches de  $X$  en utilisant la fonction de calcul de distance  $d$
3. Prendre les valeurs de  $y$  des  $K$  observations retenues :
  - a) Si on effectue une régression, calculer la moyenne (ou la médiane) de  $y$  retenues
  - b) Si on effectue une classification, calculer le mode de  $y$  retenues
4. Retourner la valeur calculée dans l'étape 3 comme étant la valeur qui a été prédite par K-NN pour l'observation  $X$ .

### Fin Algorithme

#### ➤ La Régression

C'est la tâche d'apprentissage d'un modèle dont les variables dépendantes (ou variables de sortie) sont continues. C'est l'un des problèmes les plus étudiés avec des applications dans presque tous les domaines et ainsi, de nombreux modèles de régression ont été proposés, tels que la régression linéaire avec régularisation, la régression à vecteurs de support, la régression polynomiale, ...etc. [28].



*Figure 15 : une relation linéaire simple entre deux variables  
(Exemple de régression)*

#### ➤ La Classification

Dans ML et les statistiques, la classification est le problème qui consiste à identifier à quel groupe de catégories (sous-populations) une nouvelle observation appartient, à partir

d'un ensemble d'apprentissages contenant des données dont l'appartenance à une catégorie est connue. Par exemple, attribuer un courrier électronique donné à la classe « spam » ou «non-spam» et attribuer un diagnostic à un patient donné en fonction des caractéristiques observées du patient (sexe, pression artérielle, présence ou non de certains symptômes,...etc.).

La classification est considérée comme un exemple d'apprentissage supervisé et de reconnaissance de formes, c'est-à-dire un apprentissage dans lequel un ensemble de formations d'observations correctement identifiées est disponible. La procédure non supervisée correspondante est connue sous le nom de Clustering. Elle consiste à regrouper des données en catégories en fonction d'une mesure de la similarité inhérente ou de la distance [29].

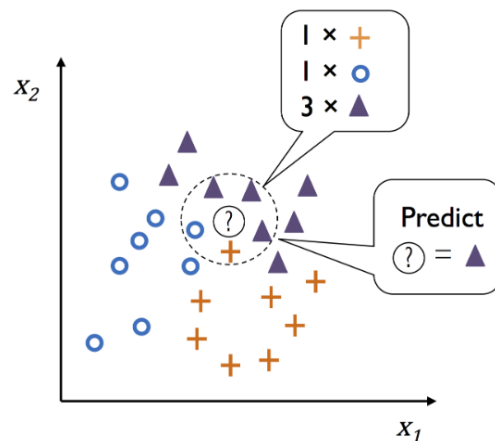


Figure 16 : illustration de KNN avec 03 classes et  $K=5$

### 2.1.2 Mesure de distance:

Afin de déterminer quels points de données sont les plus proches d'un point de requête donné, il faut calculer la distance entre le point de requête et les autres points de données. Ces mesures de distance aident à former des limites de décision, qui partitionnent les points de requête en différentes régions.

Il existe plusieurs fonctions de calcul de distance, notamment, la distance euclidienne, la distance de Manhattan, la distance de Minkowski, etc. Le choix de la fonction se fait en fonction des types de données qu'on manipule. Ainsi pour les données quantitatives (exemple : poids, salaires, taille, montant de panier électronique etc...) et du même type, la distance euclidienne est un bon candidat. Quant à la distance de Manhattan, elle est une bonne mesure à utiliser quand les données ne sont pas du même type (exemple: âge, sexe, longueur, poids etc...).

Il est inutile de coder, soi-même ces distances, généralement, les bibliothèques de ML effectuent ces calculs en interne. Il suffit juste d'indiquer la mesure de distance qu'on souhaite utiliser.[27]

On cite quelques fonctions de calculs de distance parmi les existants :

**2-1-2-1 Distance euclidienne :** Il s'agit de la mesure de distance la plus couramment utilisée, et elle est limitée aux vecteurs à valeurs réelles. En utilisant la formule ci-dessous, elle mesure une ligne droite entre le point de requête et l'autre point mesuré.[30]

**Formule :**

$$d(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2}$$

**2-1-2-2 Distance de Manhattan :** il s'agit également d'une autre mesure de distance populaire, qui mesure la valeur absolue entre deux points. Elle est également appelée taxi-distance, ou distance d'un pâté de maisons, car elle est généralement visualisée avec une grille, qui montre comment on peut naviguer d'une adresse à une autre via les rues de la ville. [30]

**Formule :**

$$d(x, y) = \left( \sum_{i=1}^n |x_i - y_i| \right)$$

**2-1-2-3 Distance de Minkowski :** Cette mesure de distance est la forme généralisée de la mesure de distance euclidienne et de celle de Manhattan. Le paramètre, p, dans la formule ci-dessous, permet la création d'autres mesures de distance. La distance euclidienne est représentée par cette formule lorsque p est égal à 2, et la distance de Manhattan est notée avec p égal à 1. [30]

**Formule :**

$$d(x, y) = \left( \sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

**2.1.3 Définir K:** La valeur  $K$  dans l'algorithme  $K$ -NN définit le nombre de voisins qui seront vérifiés pour déterminer la classification d'un point de requête spécifique. Par exemple, si  $K=1$ , l'instance sera affectée à la même classe que son seul voisin le plus proche. Définir  $K$  peut être un acte d'équilibrage, car différentes valeurs peuvent conduire à un sur-ajustement ou à un sous-ajustement. Des valeurs inférieures de  $K$  peuvent avoir un écart élevé, mais un biais faible, et des valeurs plus élevées de  $K$  peuvent entraîner un biais élevé et un écart inférieur. Le choix de  $K$  dépendra en grande partie des données d'entrée, car les données avec plus de valeurs aberrantes ou de bruit fonctionneront probablement mieux avec des valeurs de  $K$  plus élevées. Dans l'ensemble, il est recommandé d'avoir un nombre impair pour  $K$  afin d'éviter les liens de classification, et les tactiques de validation croisée peuvent aider à choisir le  $K$  optimal pour le jeu de données.[30]

**2.1.4 Application de KNN dans l'apprentissage automatique :** L'algorithme  $K$ -NN a été utilisé dans une variété d'applications, principalement dans le cadre de la classification. Certains de ces cas d'utilisation incluent :[31]

- Pré-traitement des données : les jeux de données ont souvent des valeurs manquantes, mais l'algorithme KNN peut estimer ces valeurs dans un processus connu sous le nom d'imputation des données manquantes.
- Moteurs de recommandation : en utilisant les données de parcours de navigation des sites Web, l'algorithme KNN a été utilisé pour fournir des recommandations automatiques aux utilisateurs sur du contenu supplémentaire.
- Finance : il a également été utilisé dans une variété de cas d'utilisation financière et économique. Exemple : les banques pour évaluer le risque d'un prêt à une entreprise ou à un individu.
- Médecine : KNN a également eu des applications dans l'industrie des soins de santé, en faisant des prédictions sur le risque de crises cardiaques et de cancer de la prostate. L'algorithme fonctionne en calculant les expressions géniques les plus probables.
- Reconnaissance de modèle : KNN a également aidé à identifier des modèles, tels que la classification de texte et de caractères. Cela a été particulièrement utile pour identifier les numéros manuscrits que vous pourriez trouver sur des formulaires ou des enveloppes postales.

- Agriculture: l'évaluation de la forêt inventaires et pour estimer les variables forestières. Dans ces applications, l'imagerie satellitaire est utilisée, avec l'objectif de cartographier la couverture terrestre et l'utilisation des terres avec peu de classes discrètes.
- La détection des intrusions dans les systèmes informatiques, et la gestion de bases de données d'objets en mouvement tels que des ordinateurs avec connexions sans fil.

**2.1.5 Avantages et inconvénients :** Comme tout algorithme d'apprentissage automatique, k-NN a ses forces et ses faiblesses. Selon le projet et l'application, il peut s'avérer être le bon choix ou le mauvais.

### **Avantages :**

- La facilité de mise en œuvre de cet algorithme.
- Son efficacité pour des classes réparties de manière irrégulière.
- Son efficacité pour des données incomplètes.
- La méthode des k plus proches voisins n'utilise pas de modèle pour classifier les documents.

### **Inconvénients :**

- le choix de la valeur de k (le nombre de voisins le plus proche)
- Le cout de calcul est élevé (pour chaque instance de l'ensemble de données on a besoin de calculer la distance)
- Stockage de données
- Sensible aux fonctionnalités non pertinentes

## **2.2 Algorithme Convolutional Neural Network CNN**

Les Réseaux de Neurones Convolutifs (CNNs, pour Convolutional Neural Networks) sont une catégorie puissante de modèles d'apprentissage profond spécialisés dans le traitement des données qui ont une topologie en grille, comme les images. Les CNNs sont conçus pour imiter la manière dont le cerveau humain perçoit et comprend les images en traitant les données visuelles de manière hiérarchique.

Initialement inspirés par la structure du cortex visuel des animaux, où certaines cellules sont responsables de détecter la lumière dans de petites régions du champ visuel, les CNNs exploitent une architecture similaire pour extraire progressivement des caractéristiques de plus en plus complexes des images. Cette capacité à apprendre des caractéristiques visuelles avec peu de prétraitement, contrairement aux techniques traditionnelles qui nécessitent une extraction manuelle de caractéristiques, rend les CNNs particulièrement efficaces pour des

tâches telles que la reconnaissance d'images, la classification vidéo, et même la détection d'anomalies dans les images médicales.

## 2.2.1 Les différentes couches d'un CNN

Dans un réseau de neurones convolutif (CNN), il y a plusieurs types de couches qui travaillent ensemble pour traiter et interpréter les images. Cependant, si nous devons sélectionner les quatre types de couches les plus fondamentaux et les plus couramment utilisés dans un CNN standard, nous pouvons lister les suivantes :

### 2.2.1.1 Couche de convolution

La couche de convolution est un composant fondamental des réseaux de neurones convolutifs, occupant typiquement la position de première couche. Elle est chargée de détecter un ensemble de caractéristiques dans les images fournies en entrée. Cette détection est effectuée par un processus de filtrage par convolution, où une fenêtre symbolisant la caractéristique glisse sur l'image pour calculer le produit de convolution avec chaque segment parcouru de l'image. Dans ce contexte, la caractéristique fonctionne comme un filtre, ces deux termes étant interchangeables

La convolution est une opération mathématique simple généralement utilisée pour le traitement et la reconnaissance d'images. Sur une image, son effet s'assimile à un filtrage dont voici le fonctionnement (figure 17):

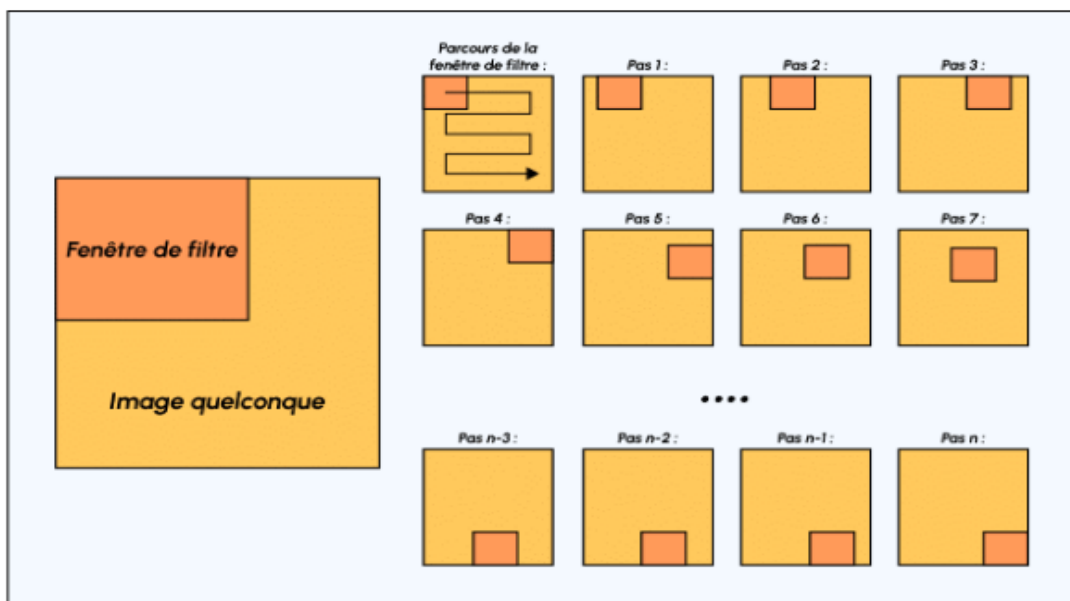


Figure17 : Schéma du parcours de la fenêtre de filtre sur l'image

- Pour commencer, nous définissons la taille de la fenêtre de filtre, située dans le coin supérieur gauche.
- Ensuite, cette fenêtre de filtre, représentant la caractéristique, se déplace de manière incrémentielle de gauche à droite par un certain nombre de cases préalablement défini (le pas), couvrant progressivement l'ensemble de l'image.
- À chaque position de la fenêtre sur l'image, un calcul de convolution est réalisé, produisant en sortie une carte d'activation ou une carte de caractéristiques.
- Cette carte indique où les caractéristiques sont localisées dans l'image : une valeur plus élevée dans la carte indique une correspondance plus forte entre la portion balayée de l'image et la caractéristique recherchée.

Pour configurer le volume de la couche de convolution, trois hyper-paramètres sont utilisés :

- **Profondeur de la couche**

Le nombre de filtres est en fait le nombre de noyaux (ou nombre de neurones associés à un même champ récepteur) choisis par couche de convolution. Le nombre de filtres déterminera la profondeur de la sortie associée à la convolution.

- **Le pas**

Lorsque l'on effectue une convolution, on choisit ce qu'on appelle le pas «stride», le pas auquel on déplace le noyau à travers l'entrant. Le pas horizontal représente le pas auquel on déplace horizontalement le noyau à travers l'image alors que le pas vertical représente le pas auquel on déplace verticalement le noyau à travers l'image. Afin de simplifier, pour la suite des choses, on utilisera un pas horizontal égal à notre pas vertical afin de simplifier le modèle

- **La marge à zéro (zero-padding)**

Parfois, il est commode de mettre des zéros à la frontière du volume d'entrée. La taille de ce 'zero-padding' est le troisième hyper paramètre. Cette marge permet de contrôler la dimension spatiale du volume de sortie. En particulier, il est parfois souhaitable de conserver la même surface que celle du volume d'entrée. Si le pas et la marge appliquée à l'image d'entrée permettent de contrôler le nombre de champs récepteurs à gérer (surface de traitement), la profondeur permet d'avoir une notion de volume de sortie, et de la même manière qu'une image peut avoir un volume, si on prend une profondeur de 3 pour les trois canaux RGB d'une image couleur, la couche de convolution va également présenter en sortie

une profondeur. C'est pour cela que l'on parle plutôt de « volume de sortie » et de « volume d'entrée », car l'entrée d'une couche de convolution peut être soit une image soit la sortie d'une autre couche de convolution.

### 2.2.1.2 Couche de pooling

Un autre concept important des CNN est le pooling, ce qui est une forme de sous échantillonnage de l'image. L'image d'entrée est découpée en une série de rectangles de  $n$  pixels de côté ne se chevauchant pas (pooling). Chaque rectangle peut être vu comme une tuile, Le signal en sortie de tuile est défini en fonction des valeurs prises par les différents pixels de la tuile. Le pooling réduit la taille spatiale d'une image intermédiaire, réduisant ainsi la quantité de paramètres et de calcul dans le réseau. Il est donc fréquent d'insérer périodiquement une couche de pooling entre deux couches convolutives successives d'une architecture CNN pour contrôler l'overfitting (sur-apprentissage). L'opération de pooling crée aussi une forme d'invariance par translation. La couche de pooling fonctionne indépendamment sur chaque tranche de profondeur de l'entrée et la redimensionne uniquement au niveau de la surface. La forme la plus courante est une couche de mise en commun avec des tuiles de taille  $2 \times 2$  (largeur/hauteur) et comme valeur de sortie la valeur maximale en entrée. On parle dans ce cas de « Max-Pool  $2 \times 2$  ». Il est possible d'utiliser d'autres fonctions de pooling que le maximum. On peut utiliser un « averagepooling » (la sortie est la moyenne des valeurs du patch d'entrée), du « L2-norm pooling ». Dans les faits, même si initialement l'average pooling était souvent utilisé il s'est avéré que le max-pooling était plus efficace car celui-ci augmente plus significativement l'importance des activations fortes. En d'autres circonstances, on pourra utiliser un pooling stochastique. Le pooling permet de gros gains en puissance de calcul. Cependant, en raison de la réduction agressive de la taille de la représentation et donc de la perte d'information associée la tendance actuelle est d'utiliser de petits filtres (type  $2 \times 2$ ). Il est aussi possible d'éviter la couche de pooling mais cela implique un risque sur-apprentissage plus important. [32]

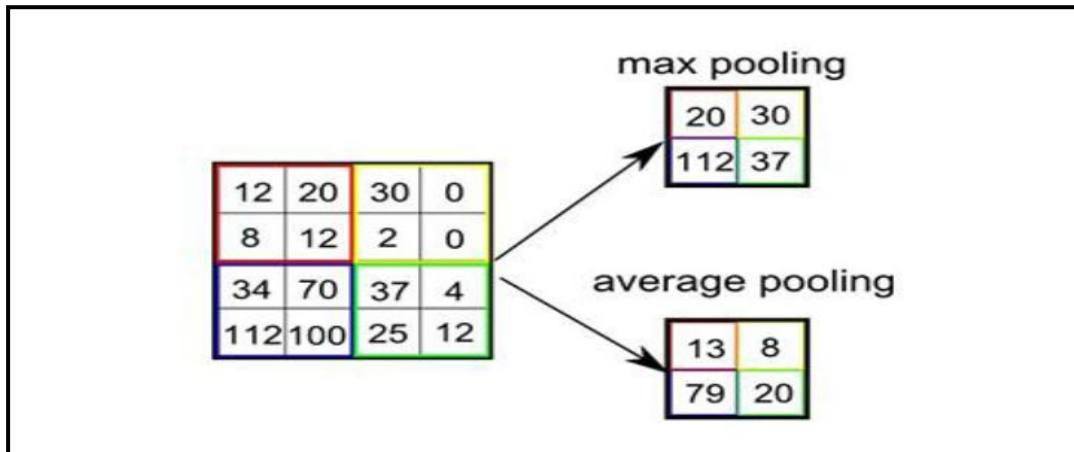


Figure18 : Pooling avec un filtre 2x2 et un pas de 2

2.2.1.3 Couche de correction ReLU

Habituellement située juste après chaque couche de convolution. La fonction ReLU, ou Rectified Linear Unit, est une fonction d'activation utilisée fréquemment dans les réseaux de neurones, notamment dans les réseaux de neurones convolutifs (CNN). Son rôle principal est d'introduire de la non-linéarité dans le modèle, ce qui est crucial car la plupart des phénomènes réels que nous tentons de modéliser ne sont pas linéaires. La fonction ReLU est définie mathématiquement comme suit :  $f(x) = \max(0, x)$ . Cela signifie que pour toutes les entrées positives, la sortie est égale à l'entrée, et pour toutes les entrées négatives, la sortie est zéro.

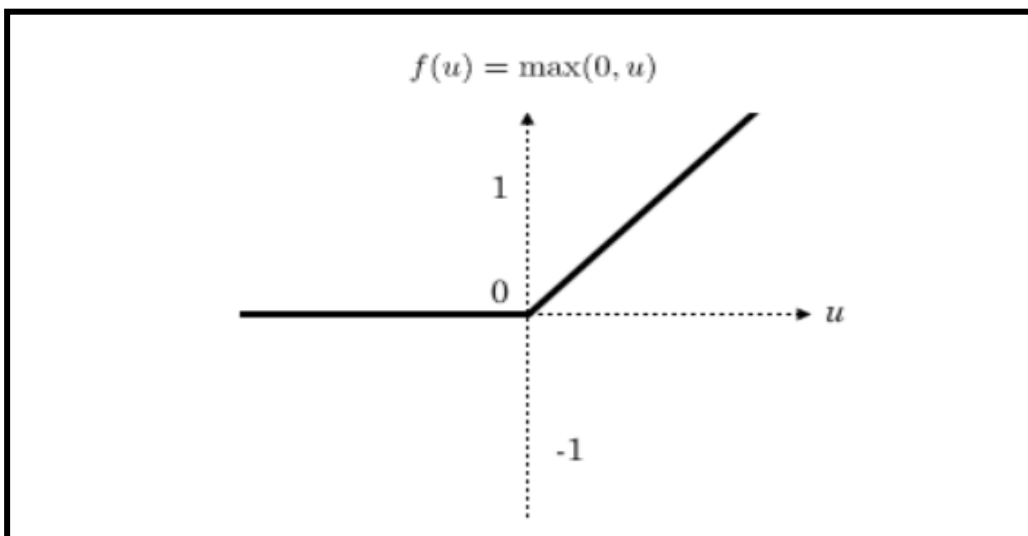


Figure 19: Schéma de la fonction d'activation ReLU

Cette simplicité confère plusieurs avantages à la fonction ReLU :

- **Accélération de la convergence** : L'utilisation de ReLU accélère souvent la convergence de l'apprentissage par descente de gradient car elle réduit le risque de disparition des gradients (vanishing gradients) lors de la propagation arrière dans les couches profondes du réseau.
- **Calcul efficace** : ReLU est une fonction très simple qui ne nécessite que des opérations de seuillage, ce qui est moins coûteux en calcul par rapport aux fonctions sigmoïde, qui nécessite des opérations exponentielles.

En résumé, dans un CNN, les couches ReLU permettent d'obtenir des représentations plus riches et non linéaires des données d'entrée, ce qui est essentiel pour apprendre des caractéristiques complexes dans des tâches telles que la reconnaissance d'image, la segmentation d'image, et autres applications de vision par ordinateur.

### 2.2.1.4 Couches Fully Connected

Ces couches sont placées en fin d'architecture de CNN et sont entièrement connectées à tous les neurones de sorties (d'où le terme fully-connected). Après avoir reçu un vecteur en entrée, la couche FC applique successivement une combinaison linéaire puis une fonction d'activation dans le but final de classifier l'input image. Elle renvoie enfin en sortie un vecteur de taille  $d$  correspondant au nombre de classes dans lequel chaque composante représente la probabilité pour l'input image d'appartenir à une classe.

### 2.2.2 Les avantages d'un CNN dans le domaine de la reconnaissance d'images

Par rapport aux réseaux neuronaux traditionnels, les réseaux de neurones à convolution (CNN) présentent plusieurs avantages significatifs [33] :

- Ils sont particulièrement bien adaptés aux applications d'apprentissage automatique et d'intelligence artificielle qui traitent de grandes quantités de données visuelles, comme la reconnaissance d'images.
- Les CNN sont robustes face aux distorsions ou autres variations optiques, ce qui les rend très fiables dans des conditions variables.
- Ces réseaux gèrent efficacement les images prises sous différents éclairages ou angles, facilitant l'identification des caractéristiques typiques des images.

- Ils nécessitent moins d'espace de stockage que les réseaux entièrement connectés, grâce à leur architecture composée de couches localement connectées. Les couches de convolution optimisent l'utilisation de l'espace de stockage.
- Le temps nécessaire pour former un CNN est également réduit, en partie grâce à l'utilisation efficace des processeurs graphiques modernes.
- Les CNN représentent la technologie de référence pour le Deep Learning et la classification dans le domaine de la reconnaissance d'images.

### Conclusion

Dans ce chapitre, nous avons exploré en détail les datasets utilisés ainsi que les algorithmes de ML sélectionnés pour élaborer un système de prédiction et de détection des incendies de forêt. Nous avons présenté deux ensembles de données distincts, chacun jouant un rôle crucial dans notre démarche. Le premier dataset se concentre sur la prédiction de magnitude et de classification des incendies, fournissant des informations telles que les conditions météorologiques, la topographie et les données historiques des incendies. Le deuxième dataset est axé sur la détection des incendies à partir d'images, permettant ainsi une approche plus réactive.

En ce qui concerne les algorithmes de ML, nous avons choisi deux approches complémentaires. Tout d'abord, l'algorithme des k plus proches voisins (KNN) a été sélectionné pour la prédiction des incendies, en se basant sur les données historiques et environnementales pour identifier les zones à risque. Ensuite, les réseaux de neurones convolutifs (CNN) ont été utilisés pour la détection en temps réel des incendies à partir d'images satellitaires, exploitant ainsi la capacité des CNNs à traiter efficacement les données visuelles.

Ce chapitre marque le début d'une approche intégrée pour lutter contre les incendies de forêt, en combinant des données variées et des techniques d'apprentissage automatique avancées. L'objectif ultime est de fournir aux gestionnaires des ressources forestières et aux autorités de secours des outils efficaces pour anticiper et répondre aux incendies de manière proactive, contribuant ainsi à la préservation des écosystèmes forestiers et à la sécurité des populations.

# **Chapitre 4**

## **Implémentation des modèles**



### Introduction

Dans ce chapitre, nous aborderons en détail l'implémentation des modèles de détection et de classification des incendies de forêt, en mettant en lumière les aspects techniques et méthodologiques de notre démarche. Ce travail repose sur deux algorithmes principaux : un réseau de neurones convolutif (CNN) pour la détection des incendies et un algorithme des k-plus proches voisins (KNN) pour la prédiction des classes d'incendies.

Nous commencerons par présenter l'environnement de travail utilisé pour le développement et la mise en œuvre de ces modèles, en décrivant le langage de programmation choisi, ainsi que les bibliothèques et outils spécifiques qui ont été employés. Ensuite, nous détaillerons le processus d'implémentation des modèles, en décrivant les différentes étapes de préparation des données, la construction des architectures des modèles, et les techniques de formation et d'optimisation utilisées.

Enfin, nous discuterons des métriques d'évaluation adoptées pour mesurer les performances de nos modèles. Ces métriques, essentielles pour l'évaluation de la précision et de l'efficacité des algorithmes de détection et de classification, permettront d'analyser les résultats obtenus et d'identifier les points forts et les éventuelles limitations de notre approche.

Ainsi, ce chapitre fournira une vision complète et détaillée de l'implémentation technique des modèles, en soulignant les choix méthodologiques et les outils technologiques employés, ainsi que les résultats de performance obtenus, afin de démontrer la pertinence et l'efficacité de notre approche dans le cadre de la détection et de la classification des incendies de forêt.

### 1- Environnement de travail

Dans cette partie, nous allons explorer l'écosystème de développement de notre modèle comprenant le langage de programmation, l'IDE et les bibliothèques utilisées.

#### 1-1 Le langage de programmation Python

Python est un langage de programmation polyvalent et très populaire, particulièrement apprécié dans les domaines de la science des données et de l'apprentissage automatique. Sa syntaxe simple et lisible, ainsi que sa vaste collection de bibliothèques et de frameworks, en font un choix privilégié pour les développeurs. Python facilite la manipulation des données, la création de modèles d'apprentissage automatique et la visualisation des résultats, ce qui le rend incontournable pour les scientifiques des données et les ingénieurs en IA.[34]

### 1-2 L'éditeur : Jupyter

Jupyter est un environnement interactif de notebook qui permet de créer et de partager des documents contenant du code, des équations, des visualisations et du texte narratif. Il est particulièrement apprécié pour l'analyse de données et le prototypage rapide de modèles. Voici quelques avantages de l'utilisation de Jupyter:[35]

- **Interactivité:** Jupyter permet d'exécuter des blocs de code indépendamment, facilitant ainsi les tests et le débogage.
- **Documentation:** Les notebooks Jupyter permettent d'intégrer des explications détaillées et des commentaires aux côtés du code, améliorant la compréhension et la reproductibilité des analyses.
- **Visualisation:** Grâce à l'intégration facile de bibliothèques comme Matplotlib, les résultats peuvent être visualisés directement dans le notebook



*Figure 20 : Logo de l'IDE Jupyter*

### 1-3 L'environnement Google Colab

Google Colab, abréviation de Google Colaboratory, est une plateforme gratuite offerte par Google qui permet d'écrire et d'exécuter du code Python directement dans un navigateur. Elle permet notamment de travailler sur des notebooks Jupyter sans se soucier du matériel ou des logiciels installés sur son ordinateur. Google Colab simplifie l'accès aux ressources de calcul et aux bibliothèques d'apprentissage automatique courantes.

Une caractéristique clé de Google Colab est son hébergement sur le cloud. Cela signifie qu'il n'est pas nécessaire d'installer Python ou d'autres bibliothèques sur votre ordinateur, car tout se fait directement dans un navigateur web. Il suffit de se connecter à son compte Google pour commencer à travailler.[36]

### 1-4 Bibliothèques

L'utilisation de bibliothèques spécialisées est très importante pour le développement efficace de modèles complexes et l'analyse de données. On va présenter les bibliothèques les plus importantes en Python pour l'apprentissage automatique et le calcul numérique.

#### 1-4-1 TensorFlow

TensorFlow est une bibliothèque open-source développée par Google pour l'apprentissage automatique et l'apprentissage profond. Elle permet de construire et d'entraîner des réseaux de neurones complexes. TensorFlow est très performant et offre une grande flexibilité grâce à son architecture basée sur les graphes de flux de données.[37]

#### 1-4-2 Scikit-Learn

Scikit-Learn est une bibliothèque simple et efficace pour l'apprentissage automatique en Python. Elle propose une vaste gamme d'outils pour l'extraction et le traitement des données, la classification, la régression, le clustering et la réduction de dimensionnalité. Sa facilité d'utilisation et son intégration fluide avec d'autres bibliothèques comme NumPy en font un choix idéal pour les débutants comme pour les experts.[38]



*Figure 21 : Logo de scikit-learn*

#### 1-4-3 NumPy

NumPy est une bibliothèque fondamentale pour le calcul numérique en Python. Elle permet de manipuler facilement des tableaux multidimensionnels et fournit une collection de fonctions mathématiques pour effectuer des opérations sur ces tableaux. NumPy est la base sur laquelle sont construites de nombreuses autres bibliothèques scientifiques en Python.[39]

### 1-4-4 Matplotlib

Matplotlib est une bibliothèque de visualisation en 2D pour Python. Elle permet de créer des graphiques statiques, animés et interactifs. Matplotlib est particulièrement utile pour visualiser des données et des résultats de manière claire et informative, ce qui est essentiel pour l'analyse et l'interprétation des données.[40]

### 1-4-5 Keras

Keras est une bibliothèque open-source qui fournit une interface haute niveau pour les réseaux de neurones, fonctionnant sur TensorFlow. Elle simplifie la création et l'entraînement de modèles d'apprentissage profond grâce à son API intuitive et conviviale. Keras est idéal pour les chercheurs et les développeurs qui souhaitent prototyper rapidement des modèles d'apprentissage profond.

### 1-4-6 Pandas

Pandas est une bibliothèque de manipulation et d'analyse de données open source en Python. Elle offre des structures de données et des outils de haut niveau pour rendre l'analyse de données simple et expressive. Pandas est largement utilisé en data science, en analyse de données, et en manipulation de données, grâce à sa capacité à gérer des données volumineuses et hétérogènes de manière efficace.[41]

## 2- Les métriques d'évaluation de modèle de classification

Les métriques d'évaluation d'un modèle de classification sont essentielles pour comprendre la performance et l'efficacité du modèle. Voici les principales métriques utilisées avec une brève description:[42], [43]

**2-1 Accuracy (Précision Globale)** :C'est le pourcentage de prédictions correctes parmi toutes les prédictions effectuées.

$$\text{Formule : Accuracy} = \frac{TN+TP}{TP+TN+FP+FN}$$

Où :

- *TP* (True Positives) : Prédits positifs et réels positifs.
- *TN* (TrueNegatives) : Prédits négatifs et réels négatifs.
- *FP* (False Positives) : Prédits positifs mais réels négatifs.
- *FN* (False Negatives) : Prédits négatifs mais réels positifs.

**Utilité :**

- Fournit une vue d'ensemble de la performance globale du modèle.
- Utile lorsque les classes sont équilibrées (c'est-à-dire que le nombre d'instances dans chaque classe est à peu près égal).

**2-2 Precision (La précision) :** Aussi appelée **précision positive**, est une métrique utilisée pour évaluer la performance d'un modèle de classification, notamment en ce qui concerne la qualité des prédictions positives. Elle mesure la proportion de vraies prédictions positives parmi toutes les prédictions positives faites par le modèle. En d'autres termes, elle indique combien de prédictions positives du modèle sont effectivement correctes.

$$\text{Formule : Precision} = \frac{TP}{TP+FP}$$

**Utilité :** La précision est particulièrement utile dans les scénarios où le coût des faux positifs est élevé. Par exemple : La détection des Spam, la détection de fraude, diagnostic médical, ...etc.

**2-3 Recall (Le rappel) :** Le rappel, également connu sous le nom de **sensibilité**, est une métrique utilisée pour évaluer la capacité d'un modèle de classification à identifier correctement les instances positives parmi toutes les instances réellement positives. En d'autres termes, le rappel mesure le taux de vrais positifs (les instances positives correctement identifiées) par rapport à l'ensemble des instances positives réelles.

$$\text{Formule : Recall} = \frac{TP}{TP+FN}$$

**Utilité :** Le rappel est particulièrement utile dans les scénarios où il est important de minimiser les faux négatifs, c'est-à-dire les instances positives manquées.

**2-4 F1 Score :** Le F1 score est une métrique utilisée pour évaluer la performance d'un modèle de classification, en combinant à la fois la précision et le rappel en une seule mesure harmonique. Il offre un équilibre entre ces deux métriques, ce qui est particulièrement utile lorsque les classes sont déséquilibrées et qu'il est important de considérer à la fois les faux positifs et les faux négatifs.

$$\text{Formule : F1 Score} = 2 \cdot \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

**Utilité :** Le F1 score est particulièrement utile dans les scénarios où il est important d'avoir un bon équilibre entre précision et rappel, surtout lorsque les classes sont déséquilibrées.

**2-5 ROC-AUC (Receiver Operating Characteristic - Area Under Curve) :** ROC-AUC est une métrique utilisée pour évaluer la performance d'un modèle de classification binaire. La courbe ROC (Receiver Operating Characteristic) trace le taux de vrais positifs (TPR) contre le taux de faux positifs (FPR) à différents seuils de classification. L'aire sous la courbe ROC (AUC - Area Under Curve) fournit une mesure unique de la capacité du modèle à discriminer entre les classes positives et négatives.

**Courbe ROC :**

- Axe des y : Taux de vrais positifs (True Positive Rate, ou Sensibilité).
- Axe des x : Taux de faux positifs (False Positive Rate).

**AUC (Area Under Curve) :**

- Représente la probabilité que le modèle classe correctement une instance positive par rapport à une instance négative.
- Varie de 0 à 1, où 1 indique une classification parfaite et 0.5 indique une classification aléatoire.

**Utilité :**

- Comparaison de Modèles : Permet de comparer plusieurs modèles de classification indépendamment du seuil de décision.
- Évaluation Globale : Offre une vue d'ensemble de la performance du modèle sur tous les seuils de classification possibles.
- Robustesse aux Classes Déséquilibrées : Moins sensible aux déséquilibres dans les classes par rapport à d'autres métriques comme la précision.
- Interprétation Facile : Un AUC plus proche de 1 indique une meilleure capacité de discrimination du modèle.

**2-6 Confusion Matrix (La matrice de confusion) :** La matrice de confusion est un outil utilisé pour évaluer la performance d'un modèle de classification. Elle permet de comparer les prédictions du modèle avec les véritables étiquettes des classes pour un ensemble de données. Chaque élément de la matrice de confusion représente le nombre de fois que des instances appartenant à des classes spécifiques ont été correctement ou incorrectement classées.

Pour un problème de classification binaire, la matrice de confusion est généralement représentée comme suit :

Prédiction Positive (P)	Prédiction Négative (N)	Prédiction Positive (P)
Classe Positive (P)	Vrai Positif (TP)	Faux Négatif (FN)
Classe Négative (N)	Faux Positif (FP)	Vrai Négatif (TN)

*Table5 : Exemple de matrice de confusion*

**Utilité** : La matrice de confusion est utile pour plusieurs raisons :

- **Évaluation Détaillée** : Elle fournit une vue détaillée des performances du modèle en termes de vrais et faux positifs/négatifs, permettant d'identifier les types d'erreurs spécifiques.
- **Calcul des Métriques** : Elle est la base pour calculer d'autres métriques de performance telles que la précision, le rappel, le F1 score, et la spécificité.
- **Analyse des Performances** : Permet de comprendre où le modèle se trompe, ce qui aide à améliorer et ajuster les modèles de classification.

**2-7 Matrice de corrélation** : La matrice de corrélation est un tableau montrant les coefficients de corrélation entre plusieurs variables. Chaque cellule de la matrice représente le coefficient de corrélation entre deux variables différentes. Ce coefficient mesure la force et la direction de la relation linéaire entre deux variables. Les valeurs de corrélation varient de -1 à 1 :

**1** indique une corrélation positive parfaite.

**-1** indique une corrélation négative parfaite.

**0** indique aucune corrélation linéaire.

**Utilité** : La matrice de corrélation est utile pour plusieurs raisons :

- **Identification des Relations** : Permet d'identifier les relations linéaires entre les variables. Cela aide à comprendre comment les variables sont liées entre elles.
- **Préparation des Données** : Aide à détecter les variables fortement corrélées avant de construire un modèle de machine learning, ce qui peut affecter les performances du modèle.
- **Analyse Exploratoire des Données** : Facilite l'analyse exploratoire en visualisant rapidement les relations entre les variables.

### 3- Implémentation et apprentissage des modèles proposés

Dans cette section, nous aborderons l'implémentation des modèles de prédiction de la magnitude des feux de forêt et de leur classification à l'aide de l'algorithme KNN, ainsi que la détection des feux grâce aux réseaux de neurones convolutifs (CNN). Nous examinerons également les paramètres spécifiques de chaque modèle et leur impact sur les performances.

### 3-1 : Le modèle de prédiction Avec KNN

#### 3-1-1 : chargement du Dataset

Le code suivant (*figure 22*) charge un fichier Excel contenant des données sur les feux de forêt, le convertit en fichier CSV, affiche les premières lignes du DataFrame pour vérifier son contenu et affiche le nombre total d'enregistrements dans le dataset.

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder

# Charger Le fichier Excel
excel_file = r'E:\Dataset\fireDataset.xlsx'
df = pd.read_excel(excel_file)

# Convertir Le DataFrame en fichier CSV
csv_file = 'fireDataset.csv'
df.to_csv(csv_file, index=False)

# Afficher L'entête du DataFrame
print("Entête du DataFrame :")
print(df.head())

# Afficher Le nombre d'enregistrements
num_records = df.shape[0]
print(f"Nombre d'enregistrements dans le dataset : {num_records}")
```

```
[5 rows x 23 columns]
Nombre d'enregistrements dans le dataset : 41132
```

*Figure 22 : Lignes de code de chargement du Dataset*

#### 3-1-2 Nettoyage du Dataset

Le code suivant (*figure 23*) charge un fichier CSV contenant des données sur les feux de forêt, nettoie les données en supprimant les lignes avec des valeurs manquantes et les doublons, puis enregistre le DataFrame nettoyé dans un nouveau fichier CSV. Il affiche également le nombre total d'enregistrements dans le DataFrame après le nettoyage et confirme l'enregistrement du fichier nettoyé.

```
# Charger Le fichier CSV
csv_file = 'fireDataset.csv'
df = pd.read_csv(csv_file)

# Supprimer Les lignes avec des valeurs manquantes
df.dropna(inplace=True)

# Supprimer Les doublons
df.drop_duplicates(inplace=True)
# Afficher Le nombre d'enregistrements
num_records = df.shape[0]
print(f"Nombre d'enregistrements dans le dataset : {num_records}")

# Enregistrer Le DataFrame nettoyé dans un nouveau fichier CSV
cleaned_csv_file = 'fireDataset_cleaned.csv'
df.to_csv(cleaned_csv_file, index=False)

print(f"Le fichier nettoyé a été enregistré sous : {cleaned_csv_file}")
```

```
Nombre d'enregistrements dans le dataset : 40900
Le fichier nettoyé a été enregistré sous : fireDataset_cleaned.csv
```

Figure 23 : Lignes de code pour le nettoyage du Dataset

### 3-1-3 conversion des valeurs non numérique

Le code suivant (*figure 24*) charge un fichier CSV nettoyé, identifie les colonnes contenant des données non numériques, et convertit ces colonnes en valeurs numériques en utilisant **LabelEncoder** de scikit-learn. Enfin, il vérifie que toutes les colonnes du DataFrame sont maintenant de type numérique en affichant les types de données des colonnes. Le dictionnaire LabelEncoders conserve les encodeurs pour chaque colonne, ce qui peut être utile pour des traitements futurs.

```
# Charger Le fichier CSV nettoyé
cleaned_csv_file = 'fireDataset_cleaned.csv'
df = pd.read_csv(cleaned_csv_file)

# Identifier Les colonnes non numériques
non_numeric_columns = df.select_dtypes(exclude=['number']).columns

# Convertir Les colonnes non numériques en valeurs numériques
label_encoders = {}
for column in non_numeric_columns:
    le = LabelEncoder()
    df[column] = le.fit_transform(df[column])
    label_encoders[column] = le

# Vérifier que toutes Les colonnes sont maintenant numériques
print("Types de données après conversion :")
print(df.dtypes)
```

Figure 24 : Lignes de code pour convertir les valeurs non numérique

### 3-1-4 La matrice de corrélation

Le code suivant (*figure 25*) calcule la matrice de corrélation des attributs numériques d'un DataFrame, puis visualise cette matrice sous forme de heatmap à l'aide de Matplotlib et Seaborn. La heatmap permet de voir facilement les relations linéaires entre les différentes paires de variables, facilitant ainsi l'analyse des données et l'identification de corrélations significatives.

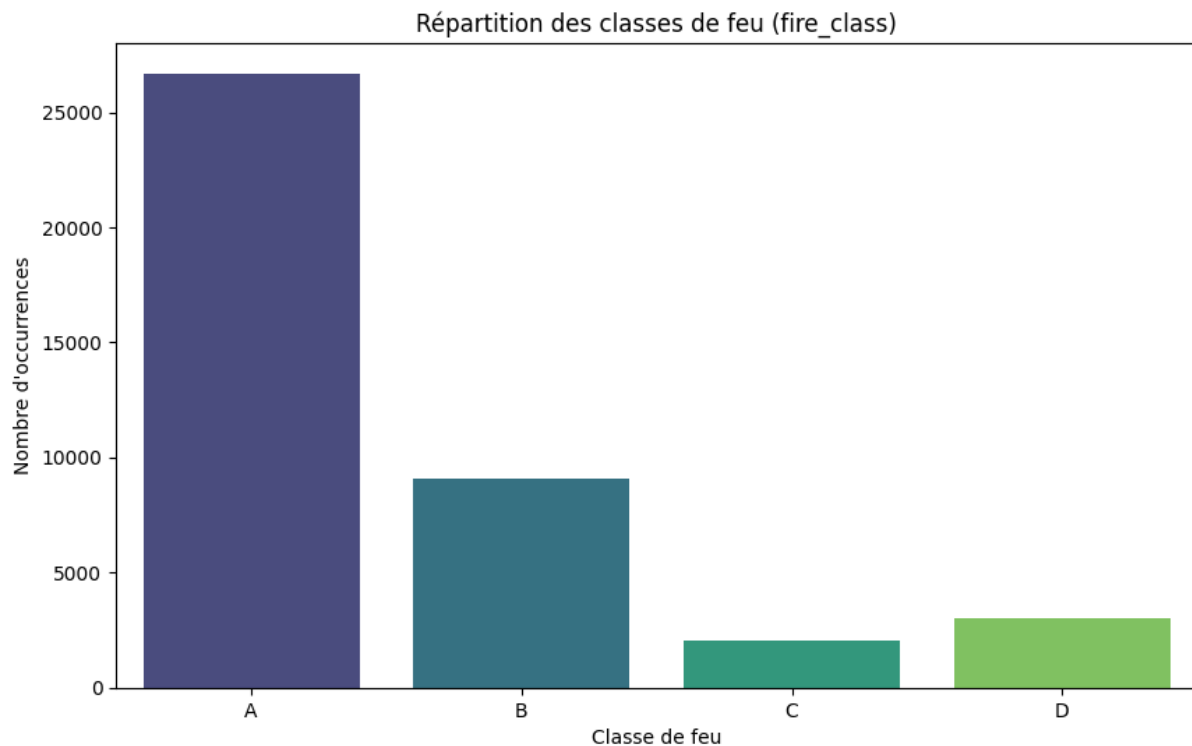
```
# Calculer la matrice de corrélation pour les attributs numériques restants
correlation_matrix = df.corr()

# Tracer la matrice de corrélation
plt.figure(figsize=(16, 12))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", square=True)
plt.title('Matrice de corrélation des attributs numériques ')
plt.show()
```

*Figure 25 : Lignes de code pour tracer la matrice de corrélation*

### 3-1-5 La répartition des classes de feu

La (*figure 26*) montre visuellement avec un histogramme la répartition des classes de feu dans notre dataset.



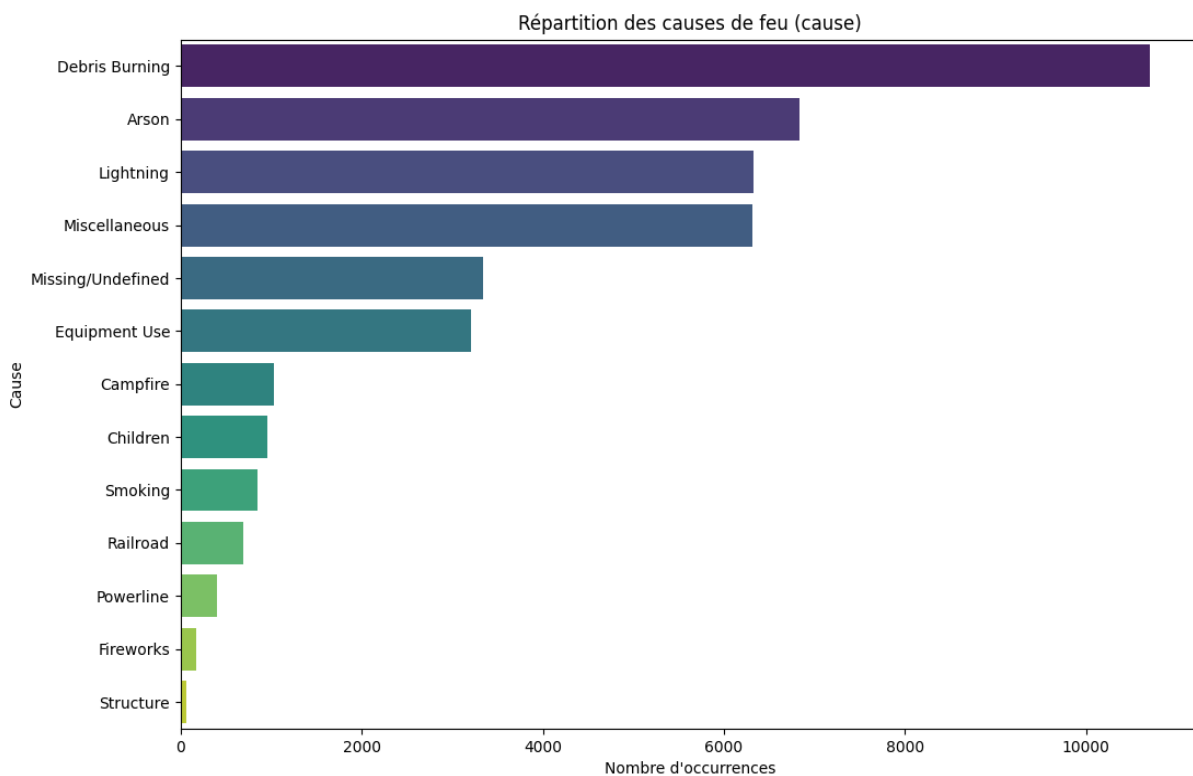
*Figure 26 : La répartition des classes de feu*

Les statistiques des classes de feu ('fire\_class') montrent la répartition des différentes classes de feu dans notre dataset :

- **Classe A** : 26,689 occurrences : Cette classe de feu est la plus fréquente dans le dataset. Cela pourrait indiquer que les feux de cette catégorie sont les plus courants dans la région ou période étudiée. La classe A pourrait représenter des feux de petite taille ou de faible intensité
- **Classe B** : 9,107 occurrences : La classe B est la deuxième classe de feu la plus courante. Cela pourrait représenter une taille ou intensité de feu moyenne.
- **Classe C** : 2,058 occurrences : Cette classe est la moins courante des quatre classes de feu
- **Classe D** : 3,046 occurrences : La classe D est moins fréquente que les classes A et B, mais encore significative. Cela pourrait représenter des feux de plus grande taille ou de plus grande intensité, survenant moins souvent.

### 3-1-6 La répartition des causes de feu

La (*figure 27*) montre la répartition des différentes causes de feu.



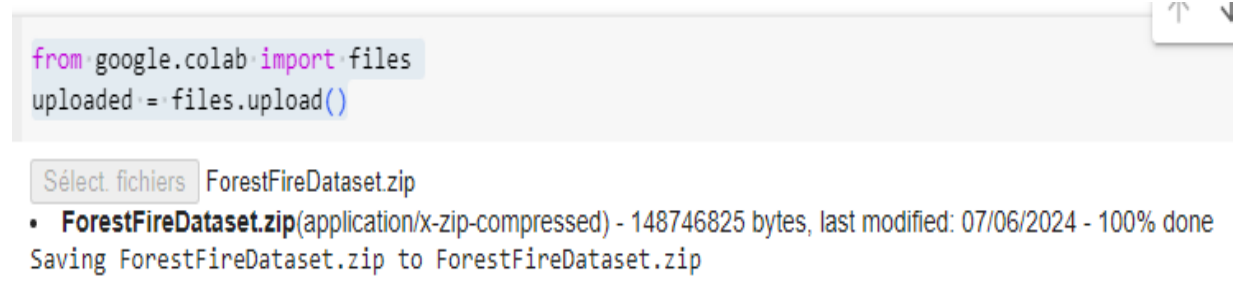
*Figure 27 : La répartition des causes de feu*

En analysant ces statistiques, on peut identifier les principales causes de feux et orienter les efforts de prévention et de gestion pour réduire l'incidence des feux dans la zone étudiée.

### 3-2 : Le modèle de détection des images de feux Avec CNN

#### 3-2-1 L'importation du Dataset

Le code ci-dessus(*figure 28*) est utilisé dans l'environnement de Google Colab pour télécharger des fichiers depuis l'appareil local vers l'environnement de Colab.



```
from google.colab import files
uploaded = files.upload()
```

Sélect fichiers ForestFireDataset.zip

- **ForestFireDataset.zip**(application/x-zip-compressed) - 148746825 bytes, last modified: 07/06/2024 - 100% done

Saving ForestFireDataset.zip to ForestFireDataset.zip

*Figure 28 : Lignes de code pour télécharger leDataset vers Google colab*

Après l'exécution de ce code, les fichiers sélectionnés sont téléchargés et stockés dans l'environnement de Google Colab, et ils peuvent être utilisés dans le notebook Colab pour diverses opérations telles que l'analyse de données, l'entraînement de modèles, etc.

#### 3-2-2 Définir le modèle

Le code suivant (*figure 29*) définit un modèle de réseau de neurones convolutionnels (CNN) pour une tâche de classification binaire sur des images. Le code définit, compile et affiche un modèle de CNN pour la classification binaire sur des images de taille 250x250 pixels avec des couches de convolution, de pooling, des couches denses, et des activations ReLU et sigmoïde. Le modèle est compilé avec l'optimiseur AdamW, la fonction de perte binaire croisée et la métrique d'exactitude.

```

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dense, Flatten, Dropout, BatchNormalization
from tensorflow.keras.optimizers import AdamW

learning_rate = 0.001

model = Sequential()
model.add(Conv2D(250, (3,3),1 , activation='relu', input_shape=(250,250,3)))
model.add(MaxPooling2D())
model.add(Conv2D(125, (3,3), 1, activation='relu'))
model.add(MaxPooling2D())
model.add(Flatten())
model.add(Dense(125, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

optimizer = AdamW(learning_rate=learning_rate)
model.compile(optimizer=optimizer, loss=tf.losses.binary_crossentropy, metrics=['accuracy'])
model.summary()

```

*Figure 29 : Lignes de code pour création de modèle CNN*

Le résultat suivant (*figure 30*) est le résumé du modèle défini dans le code fourni.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 248, 248, 250)	7000
max_pooling2d (MaxPooling2D)	(None, 124, 124, 250)	0
conv2d_1 (Conv2D)	(None, 122, 122, 125)	281375
max_pooling2d_1 (MaxPooling2D)	(None, 61, 61, 125)	0
flatten (Flatten)	(None, 465125)	0
dense (Dense)	(None, 125)	58140750
dense_1 (Dense)	(None, 1)	126

=====  
 Total params: 58429251 (222.89 MB)  
 Trainable params: 58429251 (222.89 MB)  
 Non-trainable params: 0 (0.00 Byte)

*Figure 30 : Le résultat de l'exécution du code de création du modèle CNN*

Voici une interprétation détaillée de chaque partie du résumé :

➤ **Modèle Séquentiel** : Le modèle est défini comme séquentiel, ce qui signifie que les couches sont empilées les unes sur les autres dans l'ordre défini.

### ➤ Couches :

- Deux couches de convolution (Conv2D) avec des filtres de taille (3, 3) et une activation ReLU. La première couche a 250 filtres et la deuxième couche en a 125.
- Deux couches de pooling (MaxPooling2D) pour réduire la dimension spatiale par un facteur de 2 dans chaque dimension.
- Une couche Flatten pour aplatir les données en un vecteur unidimensionnel avant de passer à une couche Dense.
- Deux couches Dense (entièrement connectées) avec une activation ReLU pour la couche cachée et une activation sigmoïde pour la couche de sortie.

### ➤ Forme de Sortie :

- La première couche de convolution produit une sortie de forme (None, 248, 248, 250), ce qui signifie qu'elle génère 250 cartes d'attributs de taille 248x248.
- Les couches de pooling réduisent progressivement la dimension spatiale.
- La couche Flatten convertit les données en un vecteur unidimensionnel de taille 475625.
- Les couches Denses réduisent progressivement la dimensionnalité jusqu'à atteindre une sortie de taille 1 (probabilité de classe pour la classification binaire).

### ➤ Nombre de Paramètres :

- Le nombre total de paramètres du modèle est de 59,753,751.
- Parmi ces paramètres, tous sont entraînaibles (59,753,751) car le modèle est entièrement connecté et utilise des couches de convolution.

Ce résumé fournit une vue d'ensemble claire de la structure du modèle, de la forme de sortie de chaque couche, ainsi que du nombre total de paramètres à entraîner. Cela est utile pour comprendre la complexité du modèle et pour diagnostiquer d'éventuels problèmes lors de l'entraînement.

### 3-2-3 Entraînement du modèle :

Le code suivant (*figure 31*) montre comment entraîner le modèle défini (model) sur les données d'entraînement (train) pour un nombre d'époques spécifié (ici 10 époques), avec une validation sur les données de test (test). Il utilise également un callback pour enregistrer les statistiques d'entraînement dans TensorBoard .

```
logdir='logs'  
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=logdir)  
hist = model.fit(train, epochs=10, validation_data=test, callbacks=[tensorboard_callback])
```

*Figure 31 : Lignes de code pour entraîner le modèle CNN*

Voici une explication détaillée :

`model.fit()` : Cette méthode entraîne le modèle sur les données d'entraînement. Les paramètres sont :

- `train` : Les données d'entraînement.
- `epochs=10` : Le nombre d'époques d'entraînement, ici défini à 10, ce qui signifie que le modèle sera entraîné sur l'ensemble des données d'entraînement 10 fois.
- `validation_data=test` : Les données de test sont utilisées pour évaluer les performances du modèle après chaque époque d'entraînement.
- `callbacks=[tensorboard_callback]` : Les callbacks sont des fonctions spéciales appelées pendant l'entraînement du modèle. Ici, `tensorboard_callback` est un callback utilisé pour enregistrer les statistiques d'entraînement dans TensorBoard.

### 3-2-4 La sauvegarde du modèle :

Le code suivant sauvegarde le modèle entraîné sous forme de fichier Keras :

```
model.save("fireDetection.keras")
```

*Figure 32 : Ligne de code pour sauvegarder le modèle CNN*

Le modèle est sauvegardé au format Keras, qui est un format standardisé pour les modèles de deep learning enregistrés avec la bibliothèque Keras. Ce format comprend les informations sur l'architecture du modèle, les poids des neurones, les configurations d'optimiseur, les paramètres de compilation, etc.

Une fois que le modèle est sauvegardé, il peut être chargé à nouveau dans un autre script ou notebook Python en utilisant la fonction `load_model()` de Keras, comme ceci :

```
model = load_model('fireDetection.keras')
```

*Figure 33 : Ligne de code pour recharger à nouveau le modèle CNN*

## 4 – Résultats et discussions

Dans cette section, nous présentons et analysons les résultats obtenus à partir des deux modèles de prédiction et de détection des incendies développés au cours de cette étude. Les modèles ont été évalués sur la base de plusieurs métriques de performance pour assurer une évaluation complète de leur efficacité et de leur précision. Ainsi, cette section offre une vue d'ensemble des performances des modèles, met en lumière leurs points forts et faibles, et propose des pistes pour des travaux futurs dans ce domaine crucial.

### 4-1 Le modèle de prédiction des classes de feux par KNN

Le code suivant évalue les performances du modèle de classification en utilisant plusieurs métriques couramment utilisées dans l'évaluation des modèles de machine learning.

```
# Évaluer le modèle
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

print("Accuracy : {:.2f}".format(accuracy))
print("Precision : {:.2f}".format(precision))
print("F1 Score : {:.2f}".format(f1))
print("\nClassification Report :")
print(classification_report(y_test, y_pred))

print("Confusion Matrix :")
conf_matrix = confusion_matrix(y_test, y_pred)
print(conf_matrix)
```

*Figure 34 : Evaluation du modèle*

Voici une explication détaillée du code et de ses résultats :

- **accuracy\_score(y\_test, y\_pred)** : Calcule le taux de précision du modèle en comparant les étiquettes prédites (y\_pred) avec les étiquettes réelles (y\_test).
- **precision\_score(y\_test, y\_pred, average='weighted')** : Calcule la précision du modèle, qui est le rapport entre les vrais positifs et tous les éléments prédits positifs. L'argument average='weighted' est utilisé pour prendre en compte les classes déséquilibrées.
- **f1\_score(y\_test, y\_pred, average='weighted')** : Calcule le score F1 du modèle, qui est une moyenne pondérée de la précision et du rappel. Comme précédemment, average='weighted' est utilisé pour traiter les classes déséquilibrées.
- **classification\_report(y\_test, y\_pred)** : Affiche un rapport de classification détaillé, comprenant la précision, le rappel, le score F1 et le support pour chaque classe.

➤ **confusion\_matrix(y\_test, y\_pred)** : Calcule la matrice de confusion, qui est une table utilisée pour décrire la performance du modèle.

### Discussion des résultats

Les résultats obtenus (*figure 35*) à partir de l'évaluation du modèle de classification sont essentiels pour évaluer son efficacité dans la tâche de prédiction.

```
Accuracy : 0.66
Precision : 0.63
F1 Score : 0.64

Classification Report :
              precision    recall  f1-score   support

     0           0.73       0.84       0.78       5324
     1           0.32       0.24       0.28       1787
     2           0.31       0.16       0.21        430
     3           0.86       0.65       0.74        639

 accuracy                0.66       8180
 macro avg              0.56       0.47       0.50       8180
 weighted avg           0.63       0.66       0.64       8180
```

*Figure35: Les résultats obtenus*

Voici une discussion sur les résultats fournis :

**Précision Globale (Accuracy):** La précision globale mesure la proportion d'échantillons correctement classés par le modèle. Dans ce cas, l'Accuracy est de 0.66, ce qui signifie que le modèle classe correctement environ 66% des échantillons. Cela indique une performance moyenne du modèle dans l'ensemble.

**Précision (Precision):** La précision mesure la proportion de vrais positifs parmi les éléments prédits positifs. Avec une précision de 0.63, le modèle a une capacité modérée à identifier correctement les échantillons positifs. Cependant, il y a une marge d'amélioration pour réduire les faux positifs.

**Score F1 :** Le score F1 est une moyenne pondérée de la précision et du rappel, et il est souvent utilisé comme mesure globale de la performance d'un modèle de classification. Avec un score F1 de 0.64, le modèle montre une performance moyenne en termes d'équilibre entre la précision et le rappel.

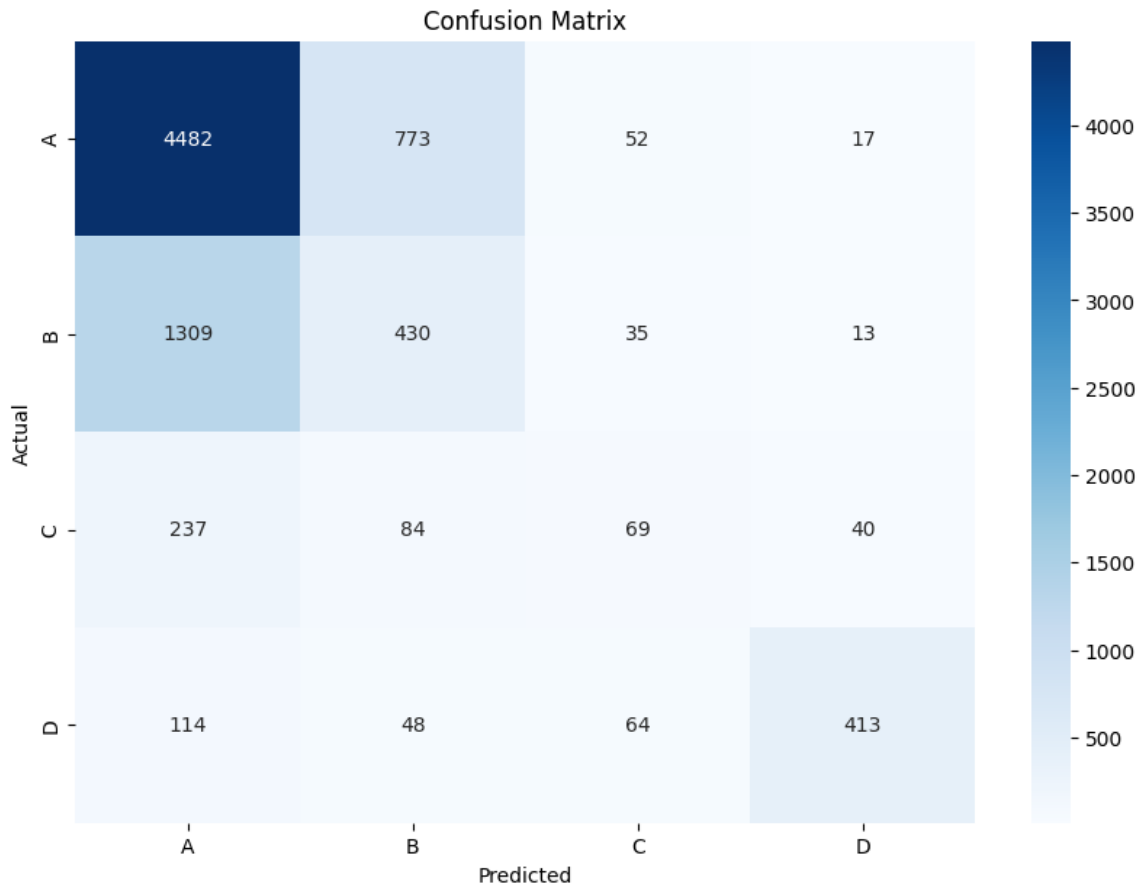
**Rapport de Classification :** Le rapport de classification fournit une analyse détaillée des performances du modèle pour chaque classe.

- **Pour la classe A**, la précision est relativement élevée à 0.73, indiquant que le modèle a une bonne capacité à prédire cette classe. Cependant, il y a un déséquilibre entre le rappel (0.84) et la précision, ce qui signifie que le modèle pourrait manquer certains échantillons de cette classe.
- **Pour la classe B**, la précision et le rappel sont plus bas, à 0.32 et 0.24 respectivement. Cela indique que le modèle a du mal à prédire correctement cette classe et peut générer un nombre significatif de faux positifs et de faux négatifs.
- **Pour la classe C**, le modèle a une précision relativement faible, ce qui signifie qu'un nombre important d'échantillons prédits comme appartenant à cette classe ne sont pas réellement de cette classe. De plus, le rappel est également faible, indiquant que le modèle manque de capacité à identifier correctement les échantillons de cette classe.
- **Pour la classe D**, le modèle présente une précision élevée, ce qui indique que la plupart des échantillons prédits comme appartenant à cette classe sont corrects. De plus, le rappel est également relativement élevé, ce qui signifie que le modèle parvient à identifier correctement une grande partie des échantillons de cette classe.

Le rapport de classification met en évidence la nécessité d'améliorer la capacité du modèle à prédire certaines classes avec précision.

### **Matrice de Confusion :**

La matrice de confusion (*figure 36*) fournit une vue détaillée des prédictions du modèle pour chaque classe, en mettant en évidence les vrais positifs, les faux positifs, les vrais négatifs et les faux négatifs. Elle montre que le modèle a tendance à confondre certaines classes, ce qui peut être une indication des zones où le modèle nécessite des améliorations.



*Figure 36 : La matrice de confusion*

En conclusion, bien que le modèle présente une certaine capacité de prédiction, il y a des aspects à améliorer, notamment en termes de précision pour certaines classes et de capacité à gérer les déséquilibres entre les classes. Des ajustements dans l'ensemble de données, les hyper paramètres du modèle ou même le choix d'un algorithme de classification différent pourraient être envisagés pour améliorer les performances globales du modèle.

#### 4-2 Le modèle de détection des images de feux par CNN

Le code suivant (*figure 37*) évalue les performances d'un modèle de classification d'images sur un ensemble de données de test en utilisant les métriques de précision, de rappel et de précision binaire. Il charge les images de test depuis un répertoire, effectue des prédictions avec le modèle et met à jour les métriques pour chaque lot d'images. Cela permet de calculer les performances globales du modèle sur l'ensemble de test.

```
[ ] pre = Precision()  
    re = Recall()  
    acc = BinaryAccuracy()
```

```
[ ] Testing = tf.keras.utils.image_dataset_from_directory('/content/working/Testing', image_size=(250,250))
```

➔ Found 380 files belonging to 2 classes.

```
[ ] for batch in Testing.as_numpy_iterator():  
    X, y = batch  
    y_pred = model.predict(X)  
    pre.update_state(y, y_pred)  
    re.update_state(y, y_pred)  
    acc.update_state(y, y_pred)
```

*Figure 37 : Lignes de code pour calculer les métriques d'évaluation*

Voici une explication du code :

- Importation et Initialisation des Métriques : Precision(), Recall() et BinaryAccuracy.
- Chargement du Dataset :
  - **tf.keras.utils.image\_dataset\_from\_directory** : Charge un ensemble de données d'images à partir d'un répertoire. Ici, les images sont chargées depuis le répertoire '/content/working/Testing' et redimensionnées à 250x250 pixels.
  - **image\_size=(250,250)** : Spécifie la taille des images à 250x250 pixels.
- Faire boucler sur les données de test :
  - **Testing.as\_numpy\_iterator()** : Convertit l'ensemble de données de test en un itérateurNumpy, permettant d'itérer sur les lots d'images et leurs étiquettes correspondantes.
  - **x, y = batch** : Pour chaque lot dans l'ensemble de données de test, x contient les images et y les étiquettes réelles.
  - **Y\_pred = model.predict(x)** : Utilise le modèle pour prédire les étiquettes des images x.
  - **pre.update\_state(y, y\_pred)** : Met à jour la métrique de précision avec les étiquettes réelles y et les prédictions y\_pred.
  - **re.update\_state(y, y\_pred)** : Met à jour la métrique de rappel de la même manière.
  - **acc.update\_state(y, y\_pred)** : Met à jour la métrique de précision binaire de la même manière

Le code suivant (*figure 38*) affiche les résultats de trois métriques de performance pour le modèle de classification binaire.

```
[ ] print(f'Precision:{pre.result()},Recall:{re.result()},Accuracy:{acc.result()}') |  
↳ Precision:0.9128205180168152,Recall:0.9368420839309692,Accuracy:0.9236842393875122
```

*Figure 38 : les résultats des trois métriques*

## Discussion des Résultats

Les résultats obtenus sont :

➤ Précision (Precision) : 0.9128

- **Interprétation** : 91.28% des échantillons prédits positifs par le modèle sont réellement positifs. Cela signifie que le modèle a une très bonne capacité à éviter les faux positifs.

- **Implication** : Une précision élevée est particulièrement importante dans des contextes où les faux positifs sont coûteux ou indésirables. Par exemple, dans la détection de feux, cela signifie que la plupart des alertes émises par le modèle sont de véritables feux.

➤ Rappel (Recall) : 0.9368

- **Interprétation** : 93.68% des échantillons réellement positifs sont correctement identifiés par le modèle. Cela montre que le modèle est excellent pour capturer les vrais positifs, bien qu'il laisse encore 6.32% non détectés.

- **Implication** : Un rappel élevé est crucial dans des situations où manquer un vrai positif (un vrai incendie) pourrait avoir des conséquences graves. Dans le contexte de la détection d'incendies, cela signifie que le modèle détecte presque tous les feux, ce qui est vital pour la sécurité et la prévention des dommages.

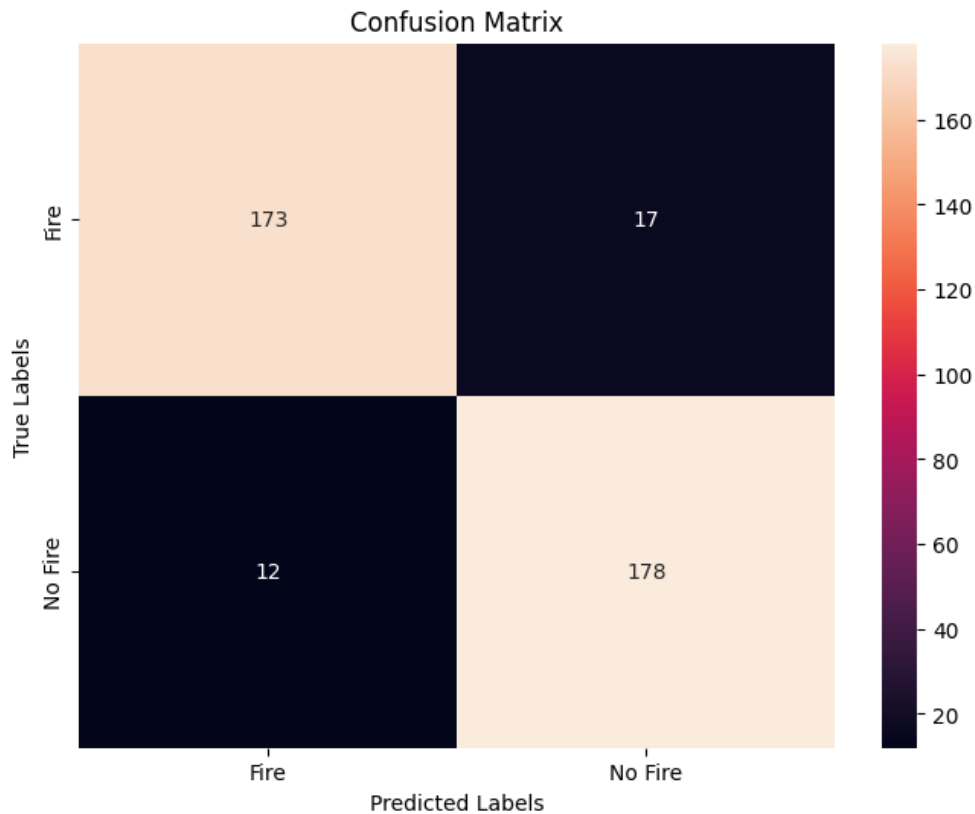
➤ Précision Binaire (Accuracy) : 0.9237

- **Interprétation** : 92.37% des échantillons totaux (positifs et négatifs) sont correctement classifiés par le modèle. Cela donne une vue d'ensemble de la performance du modèle.

- **Implication** : Une précision binaire élevée indique que le modèle performe bien globalement, mais il est important de noter que dans des ensembles de données déséquilibrés, cette métrique seule peut être trompeuse. Cependant, dans ce cas, combinée avec la haute précision et le rappel, cela indique une performance solide du modèle.

### La matrice de confusion

La matrice de confusion (*figure 39*) donne une vue détaillée des performances du modèle de classification.



*Figure 39 : Matrice de confusion pour la détection des images*

Voici une analyse des résultats affichés :

- **Axe des Y (True Labels)** : Les étiquettes réelles.
  - **Fire** : Le nombre de vrais positifs (feux correctement identifiés).
  - **No Fire** : Le nombre de vrais négatifs (non-feux correctement identifiés).
- **Axe des X (Predicted Labels)** : Les étiquettes prédites par le modèle.
  - **Fire** : Le nombre de prédictions positives (feux prédits).
  - **No Fire** : Le nombre de prédictions négatives (non-feux prédits).

### Détails des Valeurs

- **True Positives (TP)** = 173 : Le modèle a correctement identifié 173 feux.
- **False Positives (FP)** = 17 : Le modèle a incorrectement identifié 17 non-feux comme des feux.
- **False Negatives (FN)** = 12 : Le modèle a manqué 12 feux, les identifiant à tort comme des non-feux.
- **TrueNegatives (TN)** = 178 : Le modèle a correctement identifié 178 non-feux.

En conclusion, les résultats obtenus montrent que le modèle de détection d'incendies performe très bien :

- **Haute Précision** : Indique que la majorité des prédictions positives sont correctes, minimisant les fausses alertes.
- **Haut Rappel** : Indique que le modèle détecte presque tous les véritables feux, réduisant le risque de manquer des incendies potentiels.
- **Haute Précision Binaire** : Indique que le modèle a une très bonne performance globale.

Ces métriques combinées montrent que le modèle est bien équilibré et efficace pour la détection d'incendies. Il est bien adapté pour des applications où il est crucial d'identifier presque tous les incidents de feux tout en minimisant les fausses alertes. Cela est particulièrement pertinent pour des systèmes de sécurité incendie où la prévention rapide et fiable des incendies est essentielle pour protéger les biens et les vies humaines.

### Conclusion

Ce chapitre a présenté une exploration complète de l'implémentation et de l'apprentissage de modèles pour la prédiction et la détection des incendies de forêt. En utilisant le KNN pour prédire les magnitudes et les classes des incendies, ainsi que le CNN pour la détection des images d'incendies, nous avons illustré la préparation des données, le choix des environnements de travail, et l'évaluation des performances des modèles à travers diverses métriques. Les résultats obtenus montrent que les deux approches sont efficaces, avec le KNN offrant une moyenne précision pour la prédiction des classes et le CNN montre une excellente capacité à détecter les incendies dans les images. Cette étude met en évidence l'importance des choix méthodologiques et des outils utilisés pour développer des solutions robustes et précises dans la gestion des risques d'incendie, fournissant ainsi des bases solides pour de futures améliorations et applications pratiques.

# **Conclusion générale**



# Conclusion générale

---

## Conclusion générale

Notre projet de fin d'études a été une exploration passionnante de deux domaines interconnectés et cruciaux : les incendies de forêt et l'intelligence artificielle (IA). Dans le premier chapitre, nous avons examiné en profondeur la menace croissante que représentent les incendies de forêt, mettant en lumière les défis et les conséquences dévastatrices de ces catastrophes naturelles. Nous avons souligné la nécessité pressante de développer des approches préventives et de détection précoce pour atténuer leur impact sur nos écosystèmes et nos communautés.

Dans le deuxième chapitre, nous avons plongé dans le monde de l'IA, explorant ses principes fondamentaux et son potentiel révolutionnaire dans la résolution de problèmes complexes tels que la prévention des incendies de forêt. Nous avons mis en évidence les capacités des algorithmes d'apprentissage automatique à analyser des données complexes et à fournir des solutions innovantes pour anticiper et gérer les incendies de manière proactive.

Le troisième chapitre a été consacré à l'examen des bases de données cruciales utilisées dans notre projet. Nous avons décrit en détail deux ensembles de données essentiels : une base de données historique contenant des informations sur les incendies survenus aux États-Unis, ainsi qu'un ensemble d'images utilisées pour l'entraînement de nos modèles. Nous avons expliqué comment ces données ont été sélectionnées et préparées pour être utilisées dans nos algorithmes d'IA.

Dans le quatrième chapitre, nous avons présenté notre approche d'implémentation des modèles, mettant en avant le choix stratégique des algorithmes KNN et CNN. Nous avons détaillé les étapes de développement, d'entraînement et d'évaluation de ces modèles, ainsi que les résultats obtenus. Nous avons également initié une discussion sur les implications de nos résultats, soulignant l'importance de notre système dans le renforcement de la résilience face aux incendies de forêt et ouvrant la voie à de futures innovations technologiques dans ce domaine crucial.

En conclusion, notre projet représente une avancée significative dans la lutte contre les incendies de forêt, en combinant les connaissances sur les incendies eux-mêmes avec les capacités de l'IA pour fournir des solutions préventives et réactives. Nous sommes convaincus que notre travail contribuera à protéger nos écosystèmes, nos infrastructures et nos communautés contre cette menace croissante, et nous sommes enthousiasmés par les possibilités futures de développement et d'application de notre système.

# Références et bibliographies



## Références et bibliographies

---

- [1] Quintiere, J.G. (1998). *Fundamentals of Fire Phenomena*. John Wiley & Sons.
- [2] Wade, C., Dagenais, C., Klippel, M., Mikkola, E., & Werther, N. (2022). Fire dynamics. In *Fire Safe Use of Wood in Buildings* .
- [3] Quintiere, J. G. (2016). *Principles of fire behavior*. CRC Press.
- [4] Newton, A. (2007). *Forest ecology and conservation: a handbook of techniques*. Oxford University Press, USA..
- [5] Harrison, S. P., Marlon, J. R., & Bartlein, P. J. (2010). *Fire in the Earth system* (pp. 21-48). Springer Netherlands.
- [6] Alley, R., Berntsen, T., Bindoff, N. L., Chen, Z., Chidthaisong, A., Friedlingstein, P., ... & Wratt, D. (2007). Climate change 2007: The physical science basis. *Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change. Summary for Policymakers. IPCC Secretariat, Geneva, Switzerland. 21p, 980.*
- [7] González-Olabarria, J. R., Brotons, L., & Castro, J. (2008). Fire severity and seedling establishment in *Pinus halepensis* woodlands, effects of fire prevention practices. *Forest Ecology and Management*, 256(3), 428-434.
- [8] Turner, Monica G., et al. "Free-Air Carbon Dioxide Enrichment and Soil Nutrient Effects on Four North American Grasslands." *Oecologia* 112.2 (1997): 445- 448.
- [9] Bowman, David M. J. S., et al. "Fire in the Earth System." *Science* 324.5926 (2009): 481-484.
- [10] Johnston, F. H., Henderson, S. B., Chen, Y., Randerson, J. T., Marlier, M., DeFries, R. S., ... & Brauer, M. (2012). Estimated global mortality attributable to smoke from landscape fires. *Environmental health perspectives*, 120(5), 695-701.
- [11] Handmer, John W., et al. "Global Trends in the Costs of Catastrophes: Implications for National and International Policy." *Global Environmental Change* 21.3 (2011): 969-980.
- [12] Certini, G. (2005). Effects of fire on properties of forest soils: a review. *Oecologia*, 143, 1-10.
- [13] <https://www.aa.com.tr/fr/monde/en-dix-ans-la-terre-a-perdu-82-millions-d-hectares-de-foret>.
- [14] <https://www.aljazeera.net/encyclopedia/2022/7/25>.
- [15] McCarthy, J. What is artificial intelligence?.[sn]: Stanford University. *Computer Science Department, nov./2004. Disponível em: < http://wwwformal. stanford.edu/jmc/whatisai/whatisai. html>. Acesso em, 3.*

## Références et bibliographies

---

- [16] Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: a modern approach*. Pearson.
- [17] Warwick, K. (2013). *Artificial intelligence: the basics*. Routledge.
- [18] Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3), 210-229.
- [19] Azencott, C. A. (2022). *Introduction au Machine Learning-2e éd.* Dunod.
- [20] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.
- [21] Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018). A survey on deep learning for big data. *Information Fusion*, 42, 146-157.
- [22] Short, K. C. (2017). Spatial wildfire occurrence data for the United States, 1992-2015 [FPA\_FOD\_20170508].
- [23] NOAA National Centers for Environmental Information (2001): Integrated Surface Hourly [1992-2015] - <ftp://ftp.ncdc.noaa.gov/pub/data/noaa/>
- [24] Meiyappan, P., & Jain, A. K. (2012). Three distinct global estimates of historical land-cover change and land-use conversions for over 200 years. *Frontiers of earth science*, 6, 122-139.
- [25] Khan, A., Hassan, B., Khan, S., Ahmed, R., & Abuassba, A. (2022). DeepFire: a novel dataset and deep transfer learning benchmark for forest fire detection, *Mobile Inf. Syst.*
- [26] Russell, S., & Norvig, P. (2010). *Intelligence artificielle: Avec plus de 500 exercices*. Pearson Education France.
- [27] Zhang, Z. (2016). Introduction to machine learning: k-nearest neighbors. *Annals of translational medicine*, 4(11).
- [28] PanagiotisKorvesis, le 21 novembre 2017, « Apprentissage Automatique pour la Maintenance Predictive dans le Domaine de l'Aviation », Thèse de doctorat, sous la direction de MichalisVazirgiannis, Paris, de l'Université Paris-Saclay préparée à l'école Polytechnique, 139p.
- [29] Innocent Mateyaunga, 2020, « Prédictive Maintenance Using Machine Learning », la thèse de master, sous la direction de Hadj Abdelkader, Faculté de technologie de l'université de Tlemcen, 61p
- [30] Paré, M. C. (2010). *Positionnement multidimensionnel: concepts, théorie et applications*. Library and Archives Canada= Bibliothèque et Archives Canada, Ottawa, pp.137-138.

## Références et bibliographies

---

- [31] Imandoust, S. B., & Bolandraftar, M. (2013). Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background. *International journal of engineering research and applications*, 3(5), 605-610.
- [32] Francois, C. (2018). Deep learning with Python.
- [33] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
- [34] Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual.
- [35] Perkel, J. M. (2018). Why Jupyter is data scientists' computational notebook of choice. *Nature*, 563(7732), 145-147.
- [36] Bisong, E. (2019). *Building machine learning and deep learning models on Google cloud platform* (pp. 59-64). Berkeley, CA: Apress.
- [37] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). {TensorFlow}: a system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)* (pp. 265-283).
- [38] Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with Python: a guide for data scientists*. " O'Reilly Media, Inc."
- [39] Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357-362.
- [40] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in science & engineering*, 9(03), 90-95.
- [41] McKinney, W. (2010, June). Data structures for statistical computing in Python. In *SciPy* (Vol. 445, No. 1, pp. 51-56).
- [42] Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4), 427-437.
- [43] Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.