



République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la
Recherche Scientifique

ⵔⵉⵎⵓⵔ ⵉⵎⵓⵏⵉⵎ ⵉⵏⵉⵎⵓⵏⵉⵎ ⵉⵏⵉⵎⵓⵏⵉⵎ ⵉⵏⵉⵎⵓⵏⵉⵎ



Référentiel de la Formation Initiale de Troisième Cycle

« Techniques et Outils de l'IA »

2024-2025

Comité national d'encadrement, de suivi et d'exécution du programme de formation initiale du 3ème cycle dans les établissements de l'enseignement supérieur :

- M. BENYAMINA Saïd . Président de la commission nationale
- Mme. BOUALLOUCHE Rachida . Directrice de la Formation Doctorale
- M. ZEBOUCHI Mohamed Abderraouf . Sous-directeur de la Formation Doctorale
- M. BOUAROURI Djaafar . Président de la Conférence Régionale des Universités du Centre
- M. LATRECHE Mohamed El Hadi . Président de la Conférence Régionale des Universités de l'Est
- M. CHAALAL Ahmed . Président de la Conférence Régionale des Universités de l'Ouest

Comité pédagogique national de la matière « Techniques et Outils de l'Intelligence Artificielle » :

- Pr. Abdelouahab MOUSSAOUI . Président
- Pr. Baghdad ATMANI . Expert
- Pr. Saber BENHARZALLAH . Expert
- Pr. Chawki DJEDDI . Expert
- Dr. Slimane BELLAOUAR M. . Expert
- Dr. Attia NEHAR . Expert
- Dr. Omar TALBI . Expert
- Dr. Abdelhakim CHERIET . Expert
- Pr. Abderrahman YOUSFATE . Expert

1. PRÉAMBULE

1.1 Contexte et Importance de l'Intelligence Artificielle (IA)

L'intelligence artificielle (IA) est aujourd'hui au cœur de la transformation numérique et sociétale. Elle impacte de nombreux secteurs, notamment la santé, la finance, l'industrie, les sciences sociales, et la recherche scientifique. Son intégration dans les processus académiques et professionnels permet d'automatiser des tâches complexes, d'optimiser l'analyse de données massives et d'améliorer la prise de décision. En recherche scientifique, l'IA offre de nouvelles perspectives pour l'exploration et la modélisation de phénomènes complexes.

Dans le domaine universitaire, la maîtrise des techniques et outils de l'IA devient une compétence essentielle. Les doctorants doivent être capables de mobiliser ces outils pour réaliser des recherches avancées, analyser des jeux de données volumineux et rédiger des publications scientifiques de qualité. L'objectif de cette formation est de fournir une base solide en IA tout en favorisant une approche critique et éthique de son utilisation.

1.2 Enjeux et Objectifs de la Formation

L'adoption croissante des technologies basées sur l'IA s'accompagne de nombreux défis, notamment en termes de transparence, de biais algorithmiques et de protection des données. Ainsi, il est impératif de former les chercheurs aux principes fondamentaux de l'IA, à ses applications et à ses implications éthiques.

Cette formation vise à :

- **Comprendre le rôle et l'usage des outils open source dans l'IA** : Identifier les avantages des solutions open source pour le développement et l'application de l'IA, et maîtriser leur exploitation dans un cadre académique et scientifique.
- **Acquérir une maîtrise des concepts de l'IA** : Comprendre les fondements théoriques et pratiques de l'apprentissage automatique, du Deep Learning et de la science des données.
- **Développer la capacité à manipuler des outils IA avancés** : Utiliser des Framework comme TensorFlow, PyTorch et Scikit-learn pour concevoir, entraîner et évaluer des modèles intelligents.
- **Appliquer l'IA dans la recherche scientifique** : Exploiter les outils de l'IA pour l'analyse de données, la modélisation et l'extraction de connaissances dans divers domaines académiques.
- **Renforcer les compétences en traitement et visualisation de données** : Manipuler des bibliothèques comme Pandas, Matplotlib et Seaborn pour explorer et présenter des données efficacement.
- **Sensibiliser le doctorant à une utilisation maîtrisée et responsable de l'IA**, en intégrant les avancées des technologies nouvelles et les meilleures pratiques du domaine Il est essentiel de lui fournir une formation rigoureuse, lui permettant d'adapter l'IA aux exigences et aux besoins spécifiques de sa propre recherche

1.3 Intégration des Outils d'IA dans les TIC et la Recherche

L'IA vient enrichir les outils des TIC (Technologies de l'Information et de la Communication) en apportant des solutions d'automatisation, d'optimisation et d'assistance intelligente. Aujourd'hui, de nombreux outils IA facilitent la recherche et la rédaction académiques :

- **Assistance à la rédaction et à la correction** : Grammarly, LanguageTool, DeepL.
- **Vérification du plagiat et intégrité scientifique** : Turnitin, Compilatio, Plagscan.
- **Gestion des références et bibliographies** : Zotero, Mendeley, EndNote.
- **Analyse et visualisation de données** : Power BI, Tableau, Matplotlib, Seaborn.
- **Outils collaboratifs et académiques** : Google Scholar, ResearchGate, Overleaf, GitHub.

Ces outils s'intègrent dans un environnement de recherche de plus en plus numérisé, permettant aux doctorants de gagner en efficacité et en rigueur scientifique.

2. OBJECTIFS DE LA FORMATION

2.1 Comprendre les concepts de base de l'IA

- Se familiariser avec les principes fondamentaux de l'IA, incluant l'apprentissage supervisé, non supervisé et par renforcement.
- Comprendre la différence entre intelligence artificielle, apprentissage automatique et Deep Learning.
- Identifier les domaines d'application de l'IA dans la recherche scientifique et l'industrie.

2.2 Savoir prétraiter et analyser des volumes de données par des outils de l'IA

- Apprendre à collecter, nettoyer et structurer les données pour leur exploitation par des algorithmes d'IA.
- Utiliser des outils comme Pandas et Numpy pour manipuler de grands ensembles de données.
- Appliquer des techniques de réduction de dimensionnalité et de transformation des données.

2.3 Maîtriser le processus de développement de modèles d'IA

- Concevoir et implémenter des modèles d'IA avec bibliothèques R, TensorFlow, PyTorch et Scikit-learn.
- Comprendre les étapes de formation, validation et test des modèles d'IA.
- Optimiser les performances des modèles par le tuning des hyperparamètres et les techniques de validation croisée.

2.4 Comprendre et maîtriser les outils de l'IA pour la visualisation de données

- Maîtriser des bibliothèques comme Matplotlib, Seaborn et Power BI pour la représentation graphique des données.
- Apprendre à interpréter des résultats à travers des visualisations interactives.
- Gérer des Dashboard analytiques pour présenter des conclusions exploitables.

2.5 Savoir appliquer les outils de l'IA à des problèmes de recherche

- Identifier des cas d'usage de l'IA dans la recherche (santé, biologie, sciences sociales, ingénierie, etc.).
- Expérimenter avec des modèles pré-entraînés pour accélérer le développement de solutions innovantes.
- Appliquer des techniques de traitement du langage naturel (NLP) et de vision par ordinateur à des problèmes spécifiques.

2.6 Assurer une utilisation responsable et maîtrisée de l'IA et des données

- Comprendre les enjeux liés à l'utilisation de l'IA et de la science des données dans un cadre professionnel et scientifique.
- Apprendre à identifier et éviter les biais algorithmiques pour garantir des résultats fiables et équitables.
- Se conformer aux réglementations en vigueur sur la protection des données et l'utilisation responsable de l'IA

3. ORGANISATION DE LA FORMATION

3.1 Prérequis

Les doctorants doivent avoir des bases en programmation (Python), en statistiques et en traitement de données. Une connaissance des outils de bureautique avancés et de la recherche documentaire est recommandée.

3.2 Équipe de formation

La formation est assurée par :

- **Enseignants universitaires** spécialisés en IA et en science des données.
- **Chercheurs en intelligence artificielle** ayant une expertise dans l'application de l'IA à la recherche scientifique.

3.3 Formation des formateurs

Une formation des formateurs sera organisée pour assurer une homogénéité dans l'enseignement des différents modules.

3.4 Moyens Matériels et Logiciels

Afin d'assurer une formation efficace et interactive, les doctorants auront accès aux infrastructures et outils suivants :

3.4.1 Matériel

- Ordinateurs équipés de **GPU** pour l'entraînement des modèles IA.
- Accès Internet et salles équipées pour les sessions pratiques.

3.4.2 Logiciels

Les doctorants utiliseront les logiciels et environnements de développement suivants pour la programmation, l'analyse des données et la modélisation :

Catégorie	Outils
Langages de programmation	Python, R
Bibliothèques IA et Machine Learning	TensorFlow, PyTorch, Scikit-learn, Keras, PyCaret (AutoML), Caret (R), FactOminer [®] , FactoShiny [®]
Environnements de développement	Jupyter Notebook (Python), RStudio (R)
Manipulation et traitement des données	Pandas, NumPy (Python), dplyr, tidyr (R)
Visualisation et analyse de données	Matplotlib, Seaborn (Python), ggplot2, Shiny (R), Power BI, Tableau, Dash, etc.

3.5 Activités Pédagogiques Proposées pour la Matière "Techniques et Outils de l'IA"

Cette partie complète le référentiel de la matière "Techniques et Outils de l'IA" en organisant didactiquement les activités via des conférences, séminaires et ateliers, conformément aux compétences définies dans l'annexe 1 et aux objectifs du programme de l'annexe 2.

Le programme s'articule autour de trois grands axes :

AXE I : Rôle et Usage des Outils Libres et Open Source dans l'IA & la Programmation (8h)

- Introduction au Logiciels Libre et Open source (Licences GP, MIT, etc.)
- Panorama des outils open source en recherche et en enseignement.
- Exploitation des outils open source pour la gestion et la collaboration.

 Tableau – AXE I

N°	Action	Durée	Description courte	Compétences visées
1	Introduction aux concepts du logiciel libre	1h	Découvrir les principes éthiques et légaux des licences open source (GPL, MIT, Apache).	C26, C1
2	Découverte d'outils open source pour l'enseignement et la recherche	2h	Prendre en main Zotero, LibreOffice, Jupyter, R, Python, Overleaf.	C26, C13
3	Réalisation pratique de travaux scientifiques avec outils open source	2h30	S'approprier les outils essentiels : Jupyter (analyse), Zotero (bibliographie), LibreOffice (rédaction).	C13, C21
4	Gestion collaborative de projets scientifiques ouverts	2h30	Utiliser Git/GitLab, Overleaf pour la gestion de projets collaboratifs et la rédaction ouverte.	C21, C13
Total général		08 h		

AXE II : Apprentissage et communication augmentés par l'IA (8h)

- Introduction à l'IA et ses implications en apprentissage et communication.
- Assistance à la rédaction scientifique avec l'IA.
- Recherche documentaire et analyse critique assistées par l'IA.

 Tableau – AXE II

N°	Action	Durée	Description courte	Compétences visées
1	Sensibilisation aux fondements de l'IA	2h	Identifier les principes clés de l'IA et ses implications en communication scientifique.	C1, C2, C4
2	Renforcement des capacités rédactionnelles par outils intelligents	3h	Exploiter Grammarly, LanguageTool, ChatGPT, DeepL, DeepSeek, pour améliorer les publications scientifiques.	C11, C12, C13
3	Renforcement méthodologique par recherche documentaire assistée	3h	Utiliser Elicit, Semantic Scholar, Scite, Manus pour optimiser les recherches et analyses critiques des documents académiques.	C17, C13, C4
Total général		08 h		

AXE III : Recherche et analyse de données pilotées par l'IA avec Python (24)

- Introduction à la science des données avec Python.
- Prétraitement et nettoyage des données.
- Groupement, jointures et agrégations.
- Analyse exploratoire & statistiques.
- Visualisation des données.
- Modélisation prédictive & Deep Learning.
- Création d'un tableau de bord interactif.

 Tableau – AXE III

N°	Action	Durée	Description courte	Compétences visées
1	Fondamentaux en science des données	2h	Découverte de Python, Pandas et environnement data-science.	C19, C20
2	Nettoyage et structuration de données	3h	Gestion et correction des données manquantes et aberrantes.	C19, C23
3	Opérations avancées sur les données	2h	Maîtrise des fonctions groupby, merge, agg.	C20, C23
4	Analyse statistique exploratoire	2h	Moyennes, variances, tests ANOVA.	C23
5	Techniques de visualisation avancée	3h	Utilisation pratique de Seaborn et Plotly.	C25, C23
6	Techniques prédictives avancées et Deep Learning	10h	Modèles classiques ML et introduction pratique aux réseaux neuronaux avec Keras.	C19, C22, C24
7	Développement d'applications interactives	2h	Création de tableaux de bord interactifs (Streamlit, Dash).	C25, C24
Total général		24h		

AXE IV : Utilisation de l'IA pour la résolution de problèmes de recherche (8h)

- Création de tableaux de bord interactifs pour l'analyse des données
- Études de cas : application de l'IA à divers domaines
- Expérimentation sur des bases de données scientifiques

 Tableau – AXE IV

N°	Action	Durée	Description courte	Compétences visées
1	Visualisation interactive avancée des résultats scientifiques	2h	Développement d'interfaces interactives permettant une meilleure analyse des données scientifiques (Power BI, Tableau, Shiny).	C25
2	Résolution concrète de problèmes scientifiques par l'IA	3h	Implémenter des solutions IA pour des problématiques spécifiques à différents domaines (OCR, NLP, CNN).	C19, C20, C21, C22, C23, C24
3	Exploration analytique avancée de bases de données scientifiques	3h	Manipulation approfondie, prétraitement et analyse de bases de données réelles disponibles sur Kaggle ou UCI.	C19, C23, C24
Total général		08 h		

3.6. Modalités d'organisation, contenus pédagogiques et types d'évaluation

La matière « **Techniques et Outils de l'IA** » vise à fournir aux doctorants les compétences théoriques et pratiques essentielles pour intégrer efficacement les outils et les techniques avancés de l'intelligence artificielle dans leurs recherches scientifiques.

3.6.1 Types d'évaluation et organisation pédagogique :

La formation prévoit deux modalités complémentaires d'évaluation afin d'assurer une acquisition optimale des compétences visées :

- **Évaluation formative (Conférences) :**
Effectuée tout au long des conférences, elle permet d'apprécier progressivement la compréhension des concepts théoriques, de détecter les difficultés rencontrées par les doctorants et d'adapter immédiatement les démarches pédagogiques pour améliorer l'apprentissage.
- **Évaluation sommative (Ateliers et Projets pratiques) :**
Réalisée durant les ateliers et les projets pratiques, elle vise à vérifier de manière concrète l'acquisition effective des compétences en évaluant la capacité des doctorants à résoudre des problèmes scientifiques réels, à mobiliser leurs connaissances théoriques et à utiliser efficacement les outils d'intelligence artificielle dans un contexte pratique.

3.6.2 Type de formation proposée :

Conférences, Ateliers pratiques et Projets scientifiques.

3.6.3 Volume horaire global :

48 heures réparties comme suit :

- **20 heures théoriques (conférences)**
- **28 heures pratiques (ateliers et projets)**

Tableau récapitulatif du volume horaire par semestre

Semestre	Activités pédagogiques	Horaire total	Répartition détaillée
Semestre 1	Conférences	08h	Conférences 1 à 4
	Ateliers pratiques	08h	Ateliers 1 à 3
Semestre 2	Conférences	12h	Conférences 5 à 7
	Ateliers pratiques	20h	Ateliers 4 à 8
Volume horaire global	Total annuel	48h	Théoriques : 20h Pratiques : 28h

3.7. Programme des Conférences Proposées :

Les conférences sont évaluées de manière formative. Elles permettent aux doctorants d'acquérir une solide compréhension des bases théoriques et méthodologiques nécessaires à l'application efficace des outils d'IA.

Tableau détaillé des Conférences

Compétences visées	Conférences	Objectifs des conférences	Contenu de la conférence	Problèmes scientifiques traités	Durée	Évaluation
C26, C1, C13	Conférence 1 : Outils Open Source pour la recherche scientifique	Maîtriser les outils open source essentiels à la recherche scientifique	Licences (GPL, MIT), Zotero, Jupyter, Overleaf	Difficultés liées à la reproductibilité scientifique	2h	Formative
C1, C2, C4	Conférence 2 : Concepts fondamentaux de l'IA et impacts scientifiques	Comprendre les concepts fondamentaux de l'IA en contexte scientifique	Apprentissage automatique, implications scientifiques réelles	Intégration pédagogique et communication scientifique via l'IA	2h	Formative
C11, C12, C13	Conférence 3 : Rédaction scientifique assistée par l'IA	Renforcer les capacités rédactionnelles scientifiques avec l'IA	Grammarly, LanguageTool, ChatGPT, DeepL	Qualité rédactionnelle et linguistique des publications scientifiques	2h	Formative
C17, C13, C4	Conférence 4 : Recherche documentaire avancée assistée par IA	Améliorer l'analyse critique documentaire grâce à l'IA	Elicit, Semantic Scholar, Scite	Difficultés dans l'accès et la sélection de ressources scientifiques pertinentes	2h	Formative
C19, C20, C23	Conférence 5 : Analyse exploratoire et statistiques avec Python	Développer des compétences solides en analyse statistique avec Python	Python (Pandas, NumPy), statistiques descriptives et exploratoires	Traitement statistique de données expérimentales complexes	4h	Formative
C22, C24, C25	Conférence 6 : Modélisation prédictive et visualisation scientifique	Concevoir et visualiser efficacement des modèles prédictifs en recherche	Machine Learning (Scikit-learn), Deep Learning (Keras), visualisation (Plotly, Dash)	Difficultés à modéliser des données complexes en contexte scientifique	4h	Formative
C24, C25	Conférence 7 : Introduction au Deep Learning et à la modélisation avancée	Appréhender les réseaux neuronaux simples pour modéliser des phénomènes complexes	Deep Learning avec Keras (MLP de base), surapprentissage, visualisation de couches	Pourquoi et comment utiliser des réseaux neuronaux en recherche appliquée ?	4h	Formative
Volume horaire global conférences					20H	Formative

3.8. Programme des Ateliers Proposés :

Les ateliers pratiques sont évalués de manière sommative. Ils visent à valider concrètement les compétences acquises par les doctorants à travers l'application directe des outils et techniques d'IA dans des situations réelles.

Tableau détaillé des Ateliers pratiques (28h pratiques)

Compétences visées	Ateliers pratiques	Objectifs des ateliers	Activités concrètes proposées	Problèmes scientifiques traités	Durée	Évaluation
C26, C13, C21	Atelier 1 : Manipulation concrète des outils open source	Maîtriser Zotero, Jupyter, Overleaf pour la gestion efficace de documents scientifiques	Installation et utilisation pratique des outils	Difficulté à gérer et reproduire des travaux scientifiques	3h	Sommative
C21, C13	Atelier 2 : Travail collaboratif ouvert sur projets scientifiques	Maîtriser Git/GitHub et Overleaf pour la gestion collaborative	Travaux pratiques collaboratifs, gestion de versions	Gestion complexe des projets scientifiques collaboratifs	3h	Sommative
C11, C12, C13	Atelier 3 : Outils de l'IA au Service du Chercheur	Renforcer la capacité à rechercher, sélectionner et analyser des sources scientifiques à l'aide d'outils d'IA	Outils de rédaction et de recherche	Qualité rédactionnelle et linguistique des publications scientifiques	2h	Sommative
C19, C23	Atelier 4 : Prétraitement et nettoyage des données scientifiques	Préparer efficacement les données expérimentales pour l'analyse	Nettoyage et traitement avec Pandas, OpenRefine	Difficulté liée à la préparation de données expérimentales	4h	Sommative
C20, C23	Atelier 5 : Analyse statistique appliquée à des données réelles	Réaliser des analyses statistiques concrètes sur des données scientifiques	ANOVA, corrélations, tests statistiques appliqués	Complexité des analyses statistiques en recherche	4h	Sommative
C22, C24	Atelier 6 : Mise en œuvre pratique de modèles prédictifs (ML & DL)	Construire et valider des modèles prédictifs appliqués à des problématiques concrètes	Application avec Scikit-learn et Keras	Classification et prédiction de résultats scientifiques	4h	Sommative
C25, C24	Atelier 7 : Conception de tableaux de bord interactifs	Développer des visualisations scientifiques dynamiques pour la recherche	Création de dashboards avec Dash, Streamlit	Difficulté à présenter clairement des résultats scientifiques complexes	4h	Sommative
C19, C20, C21, C22, C23, C24	Atelier 8 : Projet scientifique expérimental complet avec IA	Réaliser un projet complet d'analyse, modélisation et interprétation des données	Application complète (prétraitement, analyse, modélisation, validation) sur des jeux de données réels	Difficulté d'extraire et valoriser les connaissances des grandes bases de données scientifiques	4h	Sommative
Volume horaire global ateliers					28H	Sommative

4. PROGRAMME DETAILLE

Axe 1 : Rôle et Usage des Outils Libres et Open Source dans l'IA & la Programmation(8h)

L'utilisation des outils open source est indispensable pour assurer une recherche reproductible, collaborative et accessible, notamment dans les domaines qui exploitent les technologies de l'intelligence artificielle. Cet axe permet aux doctorants de se familiariser avec ces outils essentiels et de développer une autonomie dans leur utilisation.

Thème 1 : Découvrir les principes éthiques et légaux des licences open source (GPL, MIT, Apache).

Thème 2 : Panorama des outils open source en recherche et l'enseignement.

Thème 3 : Exploitation des outils open source pour la gestion et la collaboration

1- Découvrir les principes éthiques et légaux des licences open source (GPL, MIT, Apache) (2h)

- **Objectif** : Comprendre les principes juridiques et éthiques des licences open source, ainsi que leurs implications pour la recherche scientifique et le développement logiciel.
- **Outils** : Aucun outil spécifique requis, mais analyse de documents de licence et de projets open source (ex. GitHub).
- **Contenu** :
 - *Introduction au concept de logiciel libre et open source.*
 - *Différences entre licences libres et propriétaires.*
 - *Présentation des principales licences open source :*
 - **GPL** : copyleft, réciprocité.
 - **MIT** : permissive, simple.
 - **Apache 2.0** : permissive, avec clause sur les brevets.
 - *Comparaison des licences (tableau synthétique).*
 - *Responsabilités légales et bonnes pratiques (citations, respect des clauses).*
 - *Enjeux éthiques liés à la libre diffusion des connaissances.*
-  **Atelier pratique** :
 - Étude de cas : Identifier la licence d'un projet open source sur GitHub.
 - Discussion : Quel type de licence choisir pour un projet académique ?

2- Panorama des outils open source en recherche et enseignement (2h)

- **Objectif** : Découvrir les principaux outils open source et comprendre leurs avantages pour la recherche et l'éducation.
- **Outils** : *Linux, LibreOffice, Zotero, R, Python, Jupyter Notebook*
- **Contenu** :
 - *Introduction aux logiciels libres et open source*
 - Historique et principes de l'open source.
 - Différence entre logiciels open source et propriétaires.
 - Avantages de l'open source pour la recherche scientifique (transparence, reproductibilité, collaboration).
 - *Présentation des outils open source selon leur usage (bureautique, gestion bibliographique, analyse de données).*
 - *Comparaison avec les outils propriétaires.*
 - *Enjeux éthiques et légaux*
 - Licences open source (GPL, MIT, Apache, etc.).
 - Responsabilités et bonnes pratiques d'utilisation.
 - *Le rôle de l'open source dans l'utilisation, la promotion et le développement d'intelligence artificielle.*

-  **Atelier pratique** :

Tester différents outils en fonction des besoins des doctorants (exemple : organiser une bibliographie avec Zotero, exécuter un script Python sur Jupyter).

3- Exploitation des outils open source pour la gestion et la collaboration (4h)

- **Objectif** : Utiliser des outils open source pour faciliter la gestion de projet et le travail collaboratif en recherche.
- **Outils** : *Git/GitHub, Nextcloud, Overleaf, Jitsi Meet*
- **Contenu** :
 - *Présentation des outils de gestion et de collaboration.*
 - *Gestion de version avec Git et collaboration via GitHub.*
 - *Partage et édition collaborative de documents scientifiques.*

-  **Atelier pratique** :

Travailler en équipe sur un projet de recherche en utilisant GitHub et Overleaf pour la rédaction collaborative.

Axe 2 : Apprentissage et communication augmentés par l'IA (8h)

Le deuxième axe du programme de formation vise à explorer comment l'IA peut enrichir l'apprentissage et améliorer la communication scientifique. Il est structuré en trois thèmes complémentaires, chacun apportant une approche spécifique à l'utilisation des outils d'IA dans le cadre académique.

Thème 1 : Comprendre les concepts fondamentaux de l'IA et son impact sur l'éducation et la communication.

Thème 2 : Exploiter l'IA pour améliorer la rédaction, la recherche documentaire et l'analyse critique.

Thème 3 : Développer une approche critique face aux outils d'IA pour une utilisation efficace et éthique.

Recommandation : Dans chaque thème, les outils proposés offrent des versions gratuites aux fonctionnalités parfois limitées. Il est conseillé d'explorer ces versions pour une première prise en main avant d'envisager des abonnements pour des analyses avancées. Par ailleurs, des alternatives open-source peuvent être privilégiées pour une plus grande flexibilité.

1- Introduction à l'IA et ses implications en apprentissage et communication (2h)

- **Objectif** : Ce thème a pour but de comprendre les fondements de l'IA, son évolution et ses impacts sur l'apprentissage et la communication.
- **Outils** : *ChatGPT, Claude, Gemini, Copilot*
- **Contenu** :
 - *Définition et historique de l'IA*
 - Origine et évolution des systèmes d'intelligence artificielle.
 - Différence entre l'IA symbolique, l'apprentissage automatique et les réseaux neuronaux.
 - Impact des avancées récentes sur l'éducation et la communication.
 - *Domaines d'application*
 - Éducation : tutorats intelligents, correction automatique, assistance à la rédaction.
 - Communication : synthèse et reformulation de textes, traduction automatique.
 - Production de contenu : génération d'articles, résumé automatique.
 - *Enjeux éthiques et défis*
 - Biais algorithmiques : comment les données influencent-elles les résultats ?
 - Transparence et explicabilité : pourquoi est-il important de comprendre le fonctionnement des IA ?
 - Protection des données et réglementation : bonnes pratiques et cadre légal.

-  **Atelier pratique :**

Cet atelier consistera à explorer les capacités de l'IA pour la synthèse et la reformulation de textes académiques, tout en développant un regard critique sur les productions générées.

- Prise en main des outils : utilisation de ChatGPT, Claude, Gemini, Copilot.
- Expérimentation : génération de résumés et reformulation de passages académiques.
- Analyse critique : comparaison des résultats, identification des limites et biais.
- Mise en pratique : application sur un document choisi par les doctorants.
- Retour collectif et évaluation formative : échanges sur les apprentissages et ajustements.

2- Assistance à la rédaction scientifique avec l'IA (3h)

- **Objectif :** Ce thème vise à améliorer la qualité rédactionnelle des travaux académiques en exploitant les outils d'IA pour la correction, la reformulation et l'adaptation linguistique.

- **Outils :** *Grammarly, LanguageTool, ChatGPT, DeepL*

- **Contenu :**

- *Correction et optimisation d'un texte*

- Identification des erreurs linguistiques (grammaire, orthographe, ponctuation).
- Suggestions pour améliorer la structure et la lisibilité d'un texte académique.

- *Amélioration du style et adaptation du ton rédactionnel*

- Reformulation et enrichissement du vocabulaire scientifique.
- Ajustement du niveau de langage en fonction du public cible et des exigences académiques.

- *Traduction et harmonisation multilingue*

- Utilisation des outils d'IA pour traduire et adapter des articles en plusieurs langues.
- Comparaison entre différentes solutions d'IA et leurs impacts sur la fidélité des traductions

-  **Atelier pratique :**

Cet atelier consistera à mettre en pratique les compétences acquises en utilisant les outils d'IA pour améliorer la qualité rédactionnelle d'un texte scientifique.

- Analyse et correction automatique d'un texte en utilisant Grammarly et LanguageTool.
- Reformulation et amélioration du style rédactionnel avec ChatGPT.
- Traduction et adaptation d'un passage académique dans plusieurs langues à l'aide de DeepL et d'alternatives open source.
- Comparaison des résultats obtenus entre les outils gratuits et payants.

3- Recherche documentaire et analyse critique assistées par l'IA (3h)

- **Objectif :** Ce thème vise à exploiter les capacités de l'IA pour optimiser la recherche bibliographique et améliorer l'évaluation critique des sources académiques. Il permet

aux doctorants d'accéder plus efficacement à des publications pertinentes tout en développant un regard critique sur la fiabilité des informations.

- **Outils :** *Elicit, Semantic Scholar, Scite, Zotero avec IA*
- **Contenu :**
 - *Optimisation de la recherche documentaire*
 - Utilisation des moteurs de recherche scientifiques basés sur l'IA pour identifier des articles pertinents.
 - Exploration des bases de données académiques et extraction automatisée des informations clés.
 - *Synthèse et résumé des publications*
 - Génération de résumés intelligents pour une analyse rapide des contenus académiques.
 - Comparaison des approches automatisées et manuelles dans l'évaluation des sources.
 - *Évaluation critique et validation des références*
 - Analyse des citations et des relations entre les publications pour juger de leur impact.
 - Prise en compte des biais algorithmiques et des limites des outils d'IA dans la recherche scientifique.
-  **Atelier pratique:**

L'atelier consistera à comparer une recherche documentaire classique avec une recherche assistée par IA.

- Une recherche sur un sujet académique en utilisant **Google Scholar** et une approche traditionnelle.
- Une seconde recherche avec **Elicit, Semantic Scholar et Scite**, afin de mesurer l'efficacité et les limites de l'IA dans l'analyse documentaire.
- Une discussion sur les biais algorithmiques et l'éthique dans l'usage de ces outils.

Axe 3 : Recherche et analyse de données pilotées par l'IA (24h)

L'axe II se concentre sur l'acquisition, le traitement et l'analyse des données à l'aide des outils d'intelligence artificielle (IA). L'objectif est de maîtriser les étapes fondamentales de la modélisation et de l'interprétation des résultats à travers des approches mathématiques et algorithmiques modernes.

1- Introduction à la science des données avec Python (2h)

- **Objectif :** Poser les bases techniques de la science des données avec Python. Découvrir les bibliothèques fondamentales, comprendre les structures de données et maîtriser les premières commandes d'exploration et d'inspection.
- **Outils :** *Jupyter Notebook, Google Colab, Python 3.8+, Pandas, NumPy*
- **Contenu :**

- *Chargement de données (CSV, Excel, JSON)*

```
import pandas as pd
```

```
df_csv = pd.read_csv('data.csv')
df_excel = pd.read_excel('data.xlsx')
df_json = pd.read_json('data.json')
```

- *Aperçu général du DataFrame*

```
df.head()           # 5 premières lignes
df.tail()           # 5 dernières lignes
df.sample(3)       # 3 lignes aléatoires
df.shape           # dimensions (lignes, colonnes)
df.columns         # noms des colonnes
df.dtypes          # types de chaque colonne
df.info()          # résumé complet
```

- *Statistiques de base*

```
df.describe()      # résumé statistique numérique
df['colonne'].value_counts() # fréquence des valeurs
df['colonne'].nunique() # nombre de valeurs uniques
```

- *Configuration d'affichage*

```
pd.set_option('display.max_columns', None) # afficher toutes les colonnes
pd.set_option('display.precision', 2)     # nombre de décimales
```

- *Sélection de colonnes / lignes*

```
df['marque']        # une seule colonne
df[['marque', 'prix']] # plusieurs colonnes
df.iloc[0]         # première ligne
df.iloc[0:5]       # lignes 0 à 4
df.loc[df['prix'] > 10000] # filtrage conditionnel
```

- *Création et suppression de colonnes*

```
df['prix_par_kg'] = df['prix'] / df['poids']
df.drop('ancien_colonne', axis=1, inplace=True)
```

- *Tri des valeurs*

```
df.sort_values(by='prix', ascending=False)
```

-  **Atelier :**

- Charger un fichier CSV contenant des informations sur des véhicules
- Afficher les colonnes, types, et dimensions
- Compter les valeurs uniques dans une colonne (ex. carburant)
- Créer une colonne calculée (prix_par_kg)
- Trier les voitures par prix décroissant et extraire les 10 premières

2- Prétraitement et nettoyage des données (3h)

- **Objectif :** Nettoyer, corriger, transformer les données pour garantir la qualité, la cohérence et la robustesse des analyses et des modèles prédictifs.
- **Outils :** *Pandas, NumPy, Scikit-learn* (preprocessing), OpenRefine (optionnel)
- **Contenu détaillé :**

- *Détection des valeurs manquantes*

- Localisation des *NaN* par colonne ou ligne

```
df.isnull().sum()
df[df.isnull().any(axis=1)]
```

- *Traitement des valeurs manquantes*

- Remplissage (imputation) par moyenne, médiane, constante, ou méthode avancée :

```
df['age'] = df['age'].fillna(df['age'].mean())
df['sexe'] = df['sexe'].fillna('Inconnu')
df = df.ffill() # forward fill
df = df.bfill() # backward fill
```

- Imputation avec *SimpleImputer* (*Scikit-learn*)

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='median')
df[['revenu']] = imputer.fit_transform(df[['revenu']])
```

- *Détection des valeurs aberrantes (outliers)*

- Méthode de l'écart interquartile (IQR)

```
Q1 = df['prix'].quantile(0.25)
Q3 = df['prix'].quantile(0.75)
IQR = Q3 - Q1
outliers = df[(df['prix'] < Q1 - 1.5 * IQR) | (df['prix'] > Q3 + 1.5 * IQR)]
```

- Filtrage par z-score

```
from scipy.stats import zscore
df['zscore'] = zscore(df['prix'])
df[df['zscore'].abs() > 3]
```

- *Suppression des doublons*

- Recherche et élimination des lignes répétées

```
df.duplicated().sum()
```

```
df.drop_duplicates(inplace=True)
```

- *Nettoyage des chaînes de caractères*

```
df['ville'] = df['ville'].str.strip().str.lower().str.replace('-', ' ')
```

```
df['ville'] = df['ville'].str.normalize('NFKD')
```

- *Uniformisation des types de données*

```
df['date'] = pd.to_datetime(df['date'])
```

```
df['prix'] = df['prix'].astype(float)
```

- *Création de colonnes dérivées*

- Générer de nouvelles variables utiles à partir d'existantes :

```
df['prix_par_kg'] = df['prix'] / df['poids']
```

```
df['anciennete'] = 2024 - df['annee']
```

- *Encodage des variables catégorielles*

- One-hot encoding et label encoding

```
pd.get_dummies(df, columns=['carburant'])
```

```
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()
```

```
df['sexe_code'] = le.fit_transform(df['sexe'])
```

- *Normalisation et standardisation*

- Mise à l'échelle pour les modèles sensibles (KNN, régression linéaire, etc.)

```
from sklearn.preprocessing import MinMaxScaler, StandardScaler
```

```
MinMaxScaler().fit_transform(df[['revenu']])
```

```
StandardScaler().fit_transform(df[['revenu']])
```

- *Détection des incohérences métier*

- Exemple : un poids inférieur à 300 kg ou un âge supérieur à 120 ans

```
df[df['poids'] < 300]
```

```
df[df['age'] > 120]
```

-  **Atelier pratique :**

- Identifier et traiter les valeurs manquantes, aberrantes et incohérentes
- Nettoyer des colonnes texte, unifier les formats
- Créer des colonnes utiles pour la modélisation
- Appliquer une normalisation et encoder les catégories

3- Groupement, jointures et agrégations (2h)

- **Objectif :** Organiser et résumer les données selon plusieurs axes : type de véhicule, carburant, période, etc. Obtenir des KPIs, réaliser des tableaux croisés et croiser plusieurs sources.
- **Outils :** Pandas

- **Contenu détaillé :**

- *Moyennes, sommes, comptages par catégorie*

```
df.groupby('marque')[df].mean()
df.groupby('carburant')['puissance'].sum()
df.groupby('type').size() # Nombre d'éléments par type
```

- *Agrégations multiples avec agg()*

```
df.groupby('marque').agg({
    'prix': ['mean', 'max', 'std'],
    'poids': 'median'
})
```

- *Agrégation sur plusieurs colonnes simultanées*

```
gb = df.groupby(['marque', 'carburant'])
gb['prix'].mean()
```

- *Tableaux croisés dynamiques (Pivot Tables)*

```
pd.pivot_table(df, values='prix', index='marque', columns='carburant', aggfunc='mean')
```

- *Tri de groupes agrégés*

```
df.groupby('marque')['prix'].mean().sort_values(ascending=False)
```

- *Fusions entre DataFrames (jointures relationnelles)*

```
pd.merge(df_clients, df_commandes, on='client_id', how='inner')
pd.merge(df1, df2, left_on='id', right_on='produit_id', how='outer')
```

- *Concaténation (empiler plusieurs tableaux)*

```
pd.concat([df1, df2], axis=0) # ligne par ligne
pd.concat([df1, df2], axis=1) # colonne par colonne
```

-  **Atelier :**

- Créer un tableau croisé des prix moyens par marque et type de carburant
 - Fusionner des données clients et commandes pour calculer la dépense totale par client

4- Analyse exploratoire & statistiques (2h)

- **Objectif :** Acquérir les bases statistiques pour mieux comprendre les variables d'un jeu de données : tendances centrales, dispersion, variance, corrélation simple, visualisation claire.
- **Outils :** Pandas, NumPy, Seaborn, Matplotlib, Statsmodels
- **Contenu simplifié :**

- *Statistiques de base*

- Moyenne, médiane, écart-type, quartiles

```
df.describe()
df['prix'].mean(), df['prix'].std(), df['prix'].quantile([0.25, 0.5, 0.75])
```

- *Variance et covariance*

```
df.var()
df.cov()
df.corr() # corrélation de Pearson
```

- *Visualisation des distributions*

- Histogramme et boîte à moustaches

```
import seaborn as sns
sns.histplot(df['prix'], bins=20, kde=True)
sns.boxplot(x=df['prix'])
```

- *Analyse de la variance entre groupes (ANOVA simplifiée)*

```
from scipy.stats import f_oneway
f_oneway(df[df['marque'] == 'Peugeot']['prix'], df[df['marque'] == 'Renault']['prix'])
```

- *Régression linéaire simple avec Statsmodels*

```
import statsmodels.api as sm
X = df[['poids']]
y = df['prix']
X = sm.add_constant(X) # Ajout de l'intercept
model = sm.OLS(y, X).fit()
print(model.summary())
```

-  **Atelier :**

- Comparer visuellement les distributions de prix entre marques
- Afficher les valeurs statistiques d'une variable numérique
- Vérifier les écarts-types et quartiles entre catégories

5- Visualisation des données (3h)

- **Objectif :** Représenter visuellement les relations entre variables pour faciliter l'interprétation.

- **Outils :** *Matplotlib, Seaborn, Plotly*

- *Histogramme :*

```
import seaborn as sns
sns.histplot(df['prix'], bins=30)
```

- *Nuage de points :*

```
sns.scatterplot(x='poids', y='prix', hue='marque', data=df)
```

- *Heatmap de corrélations :*

```
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
```

- *Visualisations interactives avec Plotly :*

```
import plotly.express as px
px.scatter(df, x='poids', y='prix', color='carburant')
```

-  **Atelier :**
 - Comparaison visuelle de modèles de voitures selon leur performance et prix.

6- Modélisation prédictive & Deep Learning (10h)

- **Objectif :** Construire et comparer des modèles de prédiction (classiques et simples en deep learning).
- **Outils :** *Scikit-learn, XGBoost, Keras, SHAP, Joblib, pycaret*
- **Structure du module (10h) :**
 - **1h :** Préparation et séparation du jeu de données
 - **2h :** Régression linéaire et régularisée (Lasso, Ridge)
 - **2h :** Classification (LogisticRegression, SVM, KNN)
 - **2h :** Méthodes avancées (RandomForest, XGBoost, LGBM)
 - **1h :** Introduction au Deep Learning (Keras + réseau dense)
 - **2h :** Évaluation avancée, interprétabilité (SHAP), sauvegarde

- **Contenu détaillé :**

- **1) Préparation du dataset :**

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

- **2) Modèles de régression :**

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
```

- **3) Régression par forêt aléatoire :**

```
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor()
rf.fit(X_train, y_train)
```

- **4) Régression par boosting :**

```
from xgboost import XGBRegressor
xgb = XGBRegressor()
xgb.fit(X_train, y_train)
```

- **5) Modèles de classification :**

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
clf = LogisticRegression()
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print(classification_report(y_test, y_pred))
```

- **6) Réseaux de neurones simples avec Keras :**

```
from keras.models import Sequential
from keras.layers import Dense
model = Sequential()
```

```

model.add(Dense(10, input_dim=X_train.shape[1], activation='relu'))
model.add(Dense(1, activation='linear'))
model.compile(optimizer='adam', loss='mse')
model.fit(X_train, y_train, epochs=100, batch_size=10)

```

▪ 7) *Évaluation et interprétabilité :*

```

from sklearn.metrics import mean_squared_error, r2_score
print(mean_squared_error(y_test, y_pred))
print(r2_score(y_test, y_pred))

```

▪ 8) *Explicabilité avec SHAP :*

```

import shap
explainer = shap.Explainer(model.predict, X_test)
shap_values = explainer(X_test)
shap.plots.beeswarm(shap_values)

```

▪ 9) *Utilisation AutoML avec PyCaret (Python) :*

- **Pyacaret** est une bibliothèque open-source en Python qui facilite le développement de modèles de machine Learning. C'est une alternative simplifiée à scikit-learn, conçue pour accélérer et automatiser le cycle de vie complet de la modélisation : de la préparation des données jusqu'au déploiement
- *Interface très simple : quelques lignes de code suffisent.*
- *Prétraitement automatique des données (valeurs manquantes, encodage, normalisation...).*
- *Comparaison automatique de plusieurs algorithmes.*
- *Supporte de nombreux cas d'usage :*
 - *Classification*
 - *Régression*
 - *Clustering (regroupement)*
 - *Détection d'anomalies*
 - *Analyse de texte (NLP)*
 - *Séries temporelles (forecasting)*

Exemple : Régression automatique sur un jeu de données véhicules

```

from pycaret.regression import *
import pandas as pd

# Chargement du dataset
df = pd.read_csv("voitures.csv")          # contient poids, puissance, prix, etc.

# Initialisation de l'environnement AutoML
reg = setup(data=df, target='prix', session_id=123, verbose=True)

# Comparaison automatique de tous les modèles
best_model = compare_models()

# Visualisation des résidus et de l'importance des variables
plot_model(best_model, plot='residuals')

# Visualisation de l'importance des variables
plot_model(best_model, plot='feature')

# Sauvegarde du modèle
save_model(best_model, 'modele_auto_prix')

```

-  **Atelier :**
 - Comparaison de modèles : régression linéaire vs forêt aléatoire vs réseau de neurones.
 - **Optionnel** : Reprendre quelques modèles de Machine Learning avec *PyCaret*.

- **10) Utilisation d'AutoML avec h2o.ai (R) :**
 - **h2o.ai** est une bibliothèque open-source en R qui simplifie et accélère le développement de modèles de machine learning. C'est une alternative puissante pour automatiser le cycle complet de la modélisation, de la préparation des données à la mise en production des modèles.
 - **Interface facile à utiliser** : h2o.ai permet de développer des modèles complexes avec seulement quelques lignes de code.
 - **Prétraitement automatique des données** : Gestion des valeurs manquantes, encodage des variables catégorielles, normalisation, etc.
 - **Comparaison automatique de plusieurs algorithmes** : Permet de tester différents algorithmes de machine learning pour sélectionner le plus performant.
 - **Supporte de nombreux cas d'usage** :
 - *Classification*
 - *Régression*
 - *Clustering (regroupement)*
 - *Détection d'anomalies*
 - *Analyse de texte (NLP)*
 - *Séries temporelles (forecasting)*

```
# Chargement de la bibliothèque h2o
library(h2o)

# Initialisation du serveur H2O
h2o.init()

# Chargement du dataset
df <- h2o.importFile("voitures.csv") # contient poids, puissance, prix, etc.

# Initialisation de l'environnement AutoML
aml <- h2o.automl(y = "prix", training_frame = df, max_models = 10, seed = 1234)

# Comparaison automatique de tous les modèles
leaderboard <- aml@leaderboard
print(leaderboard)

# Visualisation des performances du modèle leader
best_model <- aml@leader
performance <- h2o.performance(best_model, newdata = df)
print(performance)

# Sauvegarde du modèle
h2o.saveModel(best_model, path = "modele_auto_prix", force = TRUE)
```

-  **Atelier :**
 - **Comparaison de modèles :** Régression linéaire vs Forêt aléatoire vs Réseau de neurones avec **h2o.ai**.
 - **Optionnel :** Reprendre quelques modèles de Machine Learning avec **h2o.ai** pour observer la performance sur des jeux de données réels, comme le dataset de véhicules.

7- Création d'un tableau de bord interactif (2h)

- **Objectif :** Créer une application interactive simple utilisant uniquement Python et les bibliothèques standards pour visualiser et analyser des données. Intégration de **TensorBoard** pour l'affichage et le suivi des métriques de modèles d'apprentissage automatique.
- **Outils :** *Pandas, Matplotlib, ipywidgets, TensorBoard (via TensorFlow)*
- **Exemple détaillé :**
 - **Visualisation interactive des prix des voitures selon leur poids**

```
import pandas as pd
import matplotlib.pyplot as plt
from ipywidgets import interact, IntSlider
```

Chargement d'un jeu de données exemple

```
data = {'poids': [900, 1200, 1500, 1800, 2000], 'prix': [7000, 12000, 18000, 25000, 30000]}
df_voitures = pd.DataFrame(data)
```

Fonction interactive pour visualiser les données

```
def visualiser_prix_max(poids_max):
    df_filtre = df_voitures[df_voitures['poids'] <= poids_max]
    plt.figure(figsize=(8, 4))
    plt.scatter(df_filtre['poids'], df_filtre['prix'], color='blue')
    plt.title(f'Prix des voitures jusqu'à {poids_max} kg')
    plt.xlabel('Poids (kg)')
    plt.ylabel('Prix (€)')
    plt.grid(True)
    plt.show()
```

Widget interactif

```
interact(visualiser_prix_max, poids_max=IntSlider(min=800, max=2000, step=100, value=1500))
```

Intégration de TensorBoard pour visualiser l'entraînement d'un modèle simple

```
import tensorflow as tf
import numpy as np
```

Données simples

```
X = np.array([[0], [1], [2], [3], [4]], dtype=float)
y = np.array([[0], [1], [4], [9], [16]], dtype=float)
```

Création d'un modèle simple

```

model = tf.keras.Sequential([
    tf.keras.layers.Dense(units=10, activation='relu', input_shape=[1]),
    tf.keras.layers.Dense(units=1)
])
model.compile(optimizer='adam', loss='mse')

# Configuration de TensorBoard
log_dir = "logs/fit"
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir, histogram_freq=1)

# Entraînement avec TensorBoard
model.fit(X, y, epochs=50, callbacks=[tensorboard_callback])
    
```

Visualisation avec TensorBoard (dans un notebook Jupyter ou Colab) :

```

%load_ext tensorboard
%tensorboard --logdir logs/fit
    
```

-  **Atelier :**

- # 1. Téléchargez et chargez un fichier CSV contenant des informations réelles sur des véhicules.
- # 2. Affichez les statistiques descriptives des données chargées.
- # 3. Adaptez la fonction interactive précédente pour afficher les prix selon un autre critère, tel que la puissance ou l'année des véhicules.
- # 4. Expérimentez avec différents widgets interactifs pour enrichir l'expérience utilisateur.
- # 5. **Optionnel** : entraînez un modèle simple et affichez les résultats avec **TensorBoard**

Axe 4 : Utilisation de l'IA pour la résolution de problèmes de recherche (8h)

Cet axe vise à aider les doctorants à mettre en pratique l'intelligence artificielle à travers des études de cas adaptées à différents domaines scientifiques. Ils apprendront à maîtriser la chaîne de développement d'un processus d'IA, depuis les prétraitements des données jusqu'à l'analyse et la production de résultats, en s'appuyant sur des outils spécialisés. L'accent sera mis sur l'expérimentation avec des bases de données réelles et sur la création de tableaux de bord interactifs pour visualiser et interpréter les résultats de façon claire et pertinente. Ces apprentissages seront mis en pratique à travers des ateliers dédiés, permettant aux doctorants d'expérimenter concrètement les techniques abordées.

1- Création de tableaux de bord interactifs pour l'analyse des données (2h)

- **Objectif :** Apprendre à créer des tableaux de bord interactifs et dynamiques est essentiel pour analyser et visualiser des données scientifiques de manière efficace. Ces outils permettent une exploration approfondie des données et facilitent la communication claire des résultats, rendant les informations complexes plus accessibles et compréhensibles.
- **Outils :** *Power BI (ou Tableau), Shiny (R)*
- **Contenu :**
 - *Introduction aux tableaux de bord*
 - Présentation des outils (Power BI, Tableau, Shiny).
 - Principes de base de la visualisation des données.
 - Structuration d'un tableau de bord (objectifs, indicateurs clés, et impact sur la recherche).
 - *Création d'un tableau de bord*
 - Importation et préparation des données.
 - Création de visualisations simples (histogrammes, courbes de tendance).
 - Intégration de fonctionnalités interactives (filtres, drill-down).
 - *Analyse exploratoire*
 - Explorer les données interactivement.
 - Identifier des tendances ou des anomalies.
-  **Atelier pratique :**
 - Les doctorants créent un tableau de bord simple à partir d'un jeu de données scientifique (ex : données environnementales ou médicales).
 - Ils explorent les données et identifient des tendances ou des anomalies.

2- Études de cas : application de l'IA à divers domaines (3h)

- **Objectif** : Utiliser des techniques d'intelligence artificielle pour aborder des problèmes concrets dans divers domaines. En appliquant l'IA à des défis spécifiques, on démontre son utilité et son potentiel pour transformer la recherche, en offrant des solutions innovantes et en ouvrant de nouvelles perspectives pour résoudre des problèmes complexes.
- **Outils** : *selon le domaine (SHS ou ST) et le projet choisi.*
- **Contenu** :
 - *Présentation des cas d'usage de l'IA dans des domaines variés.*
 - *Explication des techniques d'IA adaptées au domaine d'étude.*
 - *Implémentation d'un modèle d'IA adapté au problème choisi.*
 - *Analyse des résultats et proposition d'améliorations.*
 - *Discussion sur les enjeux et les défis spécifiques de cette application.*
 - *Étude critique : forces, faiblesses et limites de l'IA dans la recherche académique.*

Exemples de projet pour les Sciences Humaines et Sociales (SHS)

- Objectif : Numériser et analyser des documents anciens (ex : lettres, journaux) pour en extraire des informations clés (dates, lieux, personnes,...).
- Techniques : OCR (reconnaissance optique de caractères), NER, classification de textes.
- Outils : Tesseract (OCR), SpaCy, Python.

Exemple de projet pour les Sciences et Technologies (ST)

- Objectif : Classifier des images médicales (ex : radiographies, IRM) pour détecter des pathologies.
- Techniques : Réseaux de neurones convolutifs (CNN), transfer learning.
- Outils : Keras, TensorFlow, OpenCV.

Exemple de projets interdisciplinaire

- Objectif : Développer un chatbot capable de répondre à des questions à la fois en SHS et en ST (ex : questions sur l'histoire des sciences et les concepts techniques).
- Techniques : Modèles de langage (GPT, BERT), traitement du langage naturel (NLP).
- Outils : Rasa, Dialogflow, Transformers.

-  **Atelier pratique** :

- **Travail en groupe sur une étude de cas** : Les doctorants appliqueront les techniques d'IA sur un problème concret.
- **Implémentation d'un modèle** : Les doctorants devront entraîner et évaluer un modèle d'IA.
- **Analyse des résultats** : Les doctorants devront interpréter les résultats et proposer des améliorations.

3- Expérimentation sur des bases de données scientifiques (3h)

- **Objectif** : Offrir aux doctorants une expérience concrète en travaillant sur des jeux de données scientifiques réels. Ils seront amenés à explorer, prétraiter et analyser ces données tout en relevant les défis spécifiques à chaque domaine. À travers cette

approche, ils apprendront à manipuler des datasets variés, à extraire des informations pertinentes et à développer une expertise en gestion et analyse de données. Ce sous-axe leur permettra ainsi de se familiariser avec des données complexes et de mieux appréhender leur utilisation dans un contexte de recherche.

- **Outils :** *Python, Pandas, NumPy, Matplotlib/Seaborn, Scikit-learn, Jupyter Notebook*
- **Contenu :**
 - *Introduction aux datasets scientifiques*
 - Présentation des types de datasets scientifiques (données tabulaires, séries temporelles, images, textes, etc.).
 - Défis spécifiques liés aux datasets scientifiques (qualité des données, volume, hétérogénéité, biais).
 - Exemples de datasets scientifiques publics (ex : UCI Machine Learning Repository, Kaggle, Open Data Portals).
 - *Exploration et prétraitement des données*
 - Techniques de nettoyage des données (gestion des valeurs manquantes, normalisation, standardisation).
 - Exploration des données (statistiques descriptives, visualisation des distributions).
 - Préparation des données pour l'analyse (feature engineering, sélection de variables).
 - *Analyse des données*
 - Application de techniques d'analyse statistique (corrélations, tests d'hypothèses).
 - Utilisation de modèles de Machine Learning simples pour l'exploration des données (ex : clustering, régression linéaire).
 - Visualisation des résultats (graphiques, cartes, etc.).
 - *Étude critique*
 - Discussion sur les limites des datasets scientifiques (qualité des données, biais potentiels, représentativité).
 - Réflexion sur l'impact des choix méthodologiques sur les résultats de recherche.

5. RÉFÉRENCES BIBLIOGRAPHIQUES

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville – [Deep Learning](#), MIT Press.
2. Aurelien Géron – *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow*, O'Reilly.
3. Stuart Russell, Peter Norvig – *Artificial Intelligence: A Modern Approach*, Pearson.
4. Ian Goodfellow – *GANs (Generative Adversarial Networks)*, ArXiv, 2014.
5. Christopher M. Bishop – *Pattern Recognition and Machine Learning*, Springer.
6. François Chollet – *Deep Learning with Python*, Manning.
7. Sebastian Raschka, Vahid Mirjalili – *Python Machine Learning*, Packt Publishing.
8. Kevin P. Murphy – *Machine Learning: A Probabilistic Perspective*, MIT Press.
9. Daphne Koller, Nir Friedman – [Probabilistic Graphical Models: Principles and Techniques](#), MIT Press.
10. Tom Mitchell – *Machine Learning*, McGraw-Hill.
11. Jure Leskovec, Anand Rajaraman, Jeffrey Ullman – [Mining of Massive Datasets](#), Cambridge University Press.
12. Michael Nielsen – *Neural Networks and Deep Learning* (available free online).
15. [Fast.ai](#) – Free courses on AI and Deep Learning.
16. [Coursera - Deep Learning Specialization](#) – Andrew Ng, Stanford.
17. Udacity - AI for Everyone – Introductory course on AI.
18. MIT OpenCourseWare – AI courses from MIT.
19. [Deep Learning AI](#) – Learning platform led by Andrew Ng.
20. [Stanford CS231n](#) – Convolutional Neural Networks for Computer Vision.
21. [Google Colab](#) – Platform for running AI models for free in the cloud.
22. [Kaggle](#) – Machine Learning competitions and practical tutorials.
23. [Hugging Face Courses](#) – Specialized courses on NLP and Transformers.
24. Google AI – Google resources and courses on AI and ML.
25. [OpenAI Blog](#) – Articles and publications on the latest advances in AI.
26. PyTorch Tutorials – Tutorials to learn PyTorch in depth.
27. [Pandas Documentation](#) – Official documentation for data analysis with Python.
28. [Seaborn Documentation](#) – Advanced Statistical Visualization.
29. [Scikit-Learn Guide](#) – Resource for predictive modeling in Python.
30. [Dash for Beginners](#): Create Interactive Data Apps with Plotly and Dash.