Algerian Democratic Republic and Populaire
Ministry of Higher Education and Scientific Research
ⵓⵖⵉⵣⵀ ⵏ ⵓⵙⵉⵎⵙⵍ ⵙⵏⵙⵅ ⵏ ⵙⵍⵓⵍⵙ ⵙⵓⵓⵍⵓⵍ

Referential for the Initial Training of Doctoral Students

« AI Techniques and Tools »

2024-2025

National Steering and Monitoring Commission:

- M. BENYAMINA Saïd . President of the National Commission
- Mme. BOUALLOUCHE Rachida . Director of Doctoral Training
- M. ZEBOUCHI Mohamed Abderraouf . Deputy Director of Doctoral Training
- M. BOUAROURI Djaafar . Pres. of the Regional Conference of Central Univ.
- M. LATRECHE Mohamed El Hadi . Pres. of the Regional Conference of Eastern Univ.
- M. CHAALAL Ahmed . Pres. of the Regional Conference of Western Univ.

The National Pedagogical Committee for Artificial Intelligence Techniques and Tools :

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

SUMMARY

1. PREAMBLE

1.1 Background and Importance of Artificial Intelligence (AI)

Artificial intelligence (AI) is now at the heart of digital and societal transformation. It impacts numerous sectors, including healthcare, finance, industry, social sciences, and scientific research. Its integration into academic and professional processes makes it possible to automate complex tasks, optimize the analysis of massive data, and improve decision-making. In scientific research, AI offers new perspectives for exploring and modeling complex phenomena.

In the academic field, mastering AI techniques and tools has become an essential skill. Doctoral students must be capable of leveraging these tools to conduct advanced research, analyze large datasets, and produce high-quality scientific publications. This program provides a strong foundation in AI while promoting a critical and ethical approach to its application.

1.2 Training Issues and Objectives

The growing adoption of AI-based technologies comes with numerous challenges, particularly concerning transparency, algorithmic bias, and data protection. Therefore, it is imperative to train researchers in the fundamental principles of AI, its applications, and its ethical implications.

This training aims to:

- **Understand the role and use of open source tools in AI:** Identify the advantages of open source solutions for AI development and application, and master their use in an academic and scientific framework.
- **Acquire a mastery of AI concepts (Gain a solid understanding of IA concepts):** Understand the theoretical and practical foundations of machine learning, deep learning, and data science.
- **Develop the ability to use (to work with) advanced AI tools:** Work with frameworks such as TensorFlow, PyTorch, and Scikit-learn to design, train, and evaluate intelligent models.
- **Applying (Apply) AI in scientific research:** Leveraging (Leverage) AI tools for data analysis, modeling, and knowledge extraction in various academic fields.
- **Enhance data processing and visualization skills:** Manipulate (Work with) libraries (such as) Pandas, Matplotlib, and Seaborn to explore and effectively present data.
- **Promote Responsible and controlled use of AI (Raise awareness of responsible and well-managed AI use):** Encourage doctoral students to integrate emerging technologies and best practices in AI, while adapting them to the specific requirements and needs of their research.

1.3 Integration of AI Tools in ICT and Research

AI enhances Information and Communication Technologies (ICT) by providing solutions for automation, optimization, and intelligent assistance. Today, many AI-powered tools facilitate academic research and writing, including:

- **Writing and proofreading assistance**: Grammarly, LanguageTool, DeepL.
- **Plagiarism checking and scientific integrity**: Turnitin, Compilatio , Plagscan .
- **Reference and bibliography management**: Zotero, Mendeley, EndNote.
- **Data analysis and visualization**: Power BI, Tableau, Matplotlib, Seaborn.
- **Collaborative and academic tools**: Google Scholar, ResearchGate, Overleaf, GitHub.

These tools fit into an increasingly digital research environment, allowing doctoral students to enhance their efficiency and scientific rigor.

---

2. TRAINING OBJECTIVES

2.1 Understand the basic concepts of AI

- Become familiar with fundamental AI principles, including supervised, unsupervised, and reinforcement learning.
- Understanding the difference between artificial intelligence, machine learning, and deep learning.
- Identify AI applications in scientific research and industry.

2.2 Learning to preprocess and analyze large data volumes using AI tools

- Learn how to collect, clean, and structure data for AI algorithms.
- Use tools like Pandas and Numpy to manipulate large data sets.
- Apply dimensionality reduction and data transformation techniques.

2.3 Mastering the process of AI model development

- Design and implement AI models with R, TensorFlow, PyTorch, and Scikit-learn libraries.
- Understand the steps involved in training, validating, and testing AI models.
- Optimize model performance through hyperparameter tuning and cross-validation techniques.

2.4 Understanding and mastering AI tools for data visualization

- Master libraries like Matplotlib, Seaborn, and Power BI for data graphical representation.
- Learn to interpret results through interactive visualizations.

- Manage analytical dashboards to present actionable findings.

2.5 Applying AI tools to research problems

- Identify use cases for AI in research fields such as healthcare, biology, social sciences, and engineering.
- Experiment with pre-trained models to accelerate the development of innovative solutions.
- Apply natural language processing (NLP) and computer vision techniques to specific problems.

2.6 Ensuring the responsible and controlled use of AI and data

- Understand the challenges of AI and data science in a professional and scientific framework.
- Learn to identify and avoid algorithmic bias to ensure reliable and fair results.
- Comply with data protection regulations and ethical guidelines for responsible AI usage.

---

3. ORGANIZATION OF TRAINING

3.1 Prerequisites

Doctoral students should have a foundational knowledge of programming (Python), statistics, and data processing. Familiarity with advanced office tools and document research is recommended.

3.2 Training team

Training is provided by:

- **University teachers** specialized in AI and data science.
- **Artificial intelligence researchers** with expertise in applying AI to scientific research.

3.3 Training of Trainers

A dedicated training session for trainers will be organized to ensure consistency in teaching across the different modules.

3.4 Hardware and Software Resources

To ensure an effective and interactive training experience, doctoral students will have access to the following infrastructure and tools:

3.4.1 Material

- Computers equipped with **GPUs** for training AI models.
- Internet access and fully equipped rooms for practical sessions.

3.4.2 Software

PhD students will use the following software and development environments for programming, data analysis and modeling:

Category	Tools
Programming languages	Python, R
AI and Machine Learning Libraries	TensorFlow, PyTorch, Scikit -learn, Keras, Caret (R), FactOminer®, FactoShiny®
Development environments	Jupyter Notebook (Python), RStudio (R)
Data handling and processing	Pandas, NumPy (Python), dplyr, tidyr (R)
Data visualization and analysis	Matplotlib, Seaborn (Python), ggplot2, Shiny (R), Power BI, Tableau, Dash, etc.

3.5 Pedagogical Activities for the Course 'Techniques and Tools of AI'

The program is structured around four main areas:

AXIS I: Role and Use of Open-Source Tools in AI

- Introduction to Free and Open-Source Software (GPL, MIT Licenses, etc.)
- Overview of open-source tools in research and education.
- Leveraging open source tools for management and collaboration.

Table – AXE I

N°	Content	Duration	Short Description	Targeted Skills
1	Introduction to Free Software Concepts	1h	Explore the ethical and legal principles of open-source licenses (GPL, MIT, Apache).	C26, C1
2	Introduction to Open-Source Tools for Teaching and Research	2h	Get hands-on experience with Zotero, LibreOffice, Jupyter, R, Python, and Overleaf.	C26, C13
3	Practical Use of Open-Source Tools for Scientific Work	2h30	Master essential tools: Jupyter (analysis), Zotero (bibliography), LibreOffice (writing).	C13, C21
4	Collaborative Management of Open Scientific Projects	2h30	Use Git/GitLab and Overleaf for collaborative project management and open writing.	C21, C13
	Total	**08 h**		

AXIS II: Learning and Communication Augmented by AI

- Introduction to AI and its impact on learning and communication.
- Scientific writing assistance with AI.
- AI-assisted document research and critical analysis.

Table – AXE II

N°	Content	Duration	Short Description	Targeted Skills
1	Awareness of AI Foundations	2h	Identify key principles of AI and its implications in scientific communication.	C1, C2, C4
2	Enhancing Writing Skills with Intelligent Tools	3h	Utilize Grammarly, LanguageTool, ChatGPT, DeepL, and DeepSeek to improve scientific publications.	C11, C12, C13
3	Methodological Enhancement through AI-Assisted Research	3h	Use Elicit, Semantic Scholar, Scite, and Manus to optimize research and critical analysis of academic documents.	C17, C13, C4
	Total	**08 h**		

AXIS III: AI-driven data research and analysis with Python

- Introduction to Data Science with Python.
- Data preprocessing and cleaning.
- Grouping, joins, and aggregations.
- Exploratory analysis & statistics.
- Data visualization.
- Predictive Modeling & Deep Learning.
- Creation of an interactive dashboard.

Table – AXE III

N°	Content	Duration	Short Description	Targerted Skills
1	Fundamentals of Data Science	2h	Introduction to Python, Pandas, and the data science environment.	C19, C20
2	Data Cleaning and Structuring	3h	Handling and correcting missing and anomalous data.	C19, C23
3	Advanced Data Operations	2h	Mastering groupby, merge, and agg functions.	C20, C23
4	Exploratory Statistical Analysis	2h	Means, variances, ANOVA tests.	C23
5	Advanced Visualization Techniques	3h	Practical use of Seaborn and Plotly.	C25, C23
6	Advanced Predictive Techniques and Deep Learning	10h	Classical ML models and a practical introduction to neural networks with Keras.	C19, C22, C24
7	Development of Interactive Applications	2h	Creating interactive dashboards (Streamlit, Dash).	C25, C24
	Total	**24h**		

AXIS IV: Use of AI for solving research problems

- Creating interactive dashboards for data analysis
- Case studies: application of AI in various fields
- Experimentation with scientific databases

Table – AXE IV

N°	Content	Duration	Short Description	Targeted Skills
1	Advanced Interactive Visualization of Scientific Results	2h	Developing interactive interfaces for better scientific data analysis (Power BI, Tableau, Shiny).	C25
2	Practical Problem-Solving in Science Using AI	3h	Implementing AI solutions for domain-specific problems (OCR, NLP, CNN).	C19, C20, C21, C22, C23, C24
3	Advanced Analytical Exploration of Scientific Databases	3h	In-depth manipulation, preprocessing, and analysis of real databases from Kaggle or UCI.	C19, C23, C24
	Total	**08 h**		

3.6. Organization Methods, Educational Content, and Assessment Types

The course 'Techniques and Tools of AI' aims to provide doctoral students with the essential theoretical and practical skills to effectively integrate advanced artificial intelligence tools and techniques into their scientific research.

3.6.1 Types of Assessment and Pedagogical Organization:

The training includes two complementary assessment methods to ensure optimal acquisition of the targeted skills:

- **Formative Assessment (Lectures):**
  Conducted throughout the lectures, this assessment progressively evaluates the understanding of theoretical concepts, identifies difficulties encountered by doctoral students, and allows for immediate adjustments to teaching approaches to enhance learning.

- **Summative Assessment (Workshops and Practical Projects):**
  Carried out during workshops and practical projects, this assessment aims to concretely verify the effective acquisition of skills by evaluating doctoral students' ability to solve real scientific problems, apply their theoretical knowledge, and efficiently use artificial intelligence tools in a practical context.

3.6.2 Type de Training Offered:

Lectures, Practical Workshops, and Scientific Projects.

3.6.3 Total Hourly Volume:

48 hours distributed as follows:

- **20 theoretical hours (lectures)**
- **28 practical hours (workshops and projects)**

Summary Table of Hourly Volume per Semester

Semester	Pedagogical Activities	Total Hours	Detailed Breakdown
Semester 1	Lectures	08h	Lectures 1 to 4
	Practical Workshops	08h	Workshops 1 to 3
Semester 2	Lectures	12h	Lectures 5 to 7
	Practical Workshops	20h	Workshops 4 to 8
Total Annual Hours		**48h**	**Theoretical : 20h** **Practical : 28h**

3.7. Program of Proposed Lectures

The lectures are evaluated formatively. They allow doctoral students to acquire a solid understanding of the theoretical and methodological foundations necessary for the effective application of AI tools.

Detailed Table of Lectures (20 Hours)

Targeted Skills	Lectures	Lecture Objectives	Lecture Content	Scientific Problems Addressed	Duration	Evaluation
C26, C1, C13	**Lecture 1**: Open Source Tools for Scientific Research	Master essential open-source tools for scientific research	Licenses (GPL, MIT), Zotero, Jupyter, Overleaf	Challenges related to scientific reproducibility	2h	Formative
C1, C2, C4	**Lecture 2**: Fundamental AI Concepts and Scientific Impact	Understand fundamental AI concepts in a scientific context	Machine learning, real scientific implications	AI-based pedagogical integration and scientific communication	2h	Formative
C11, C12, C13	**Lecture 3:** AI-Assisted Scientific Writing	Enhance scientific writing skills with AI	Grammarly, LanguageTool, ChatGPT, DeepL	Quality of scientific writing and linguistic accuracy in publications	2h	Formative
C17, C13, C4	**Lecture 4**: Advanced AI-Assisted Literature Review	Improve critical document analysis using AI	Elicit, Semantic Scholar, Scite	Challenges in accessing and selecting relevant scientific resources	2h	Formative
C19, C20, C23	**Lecture 5**: Exploratory Analysis and Statistics with Python	Develop strong statistical analysis skills with Python	Python (Pandas, NumPy), descriptive and exploratory statistics	Statistical processing of complex experimental data	4h	Formative
C22, C24, C25	**Lecture 6:** Predictive Modeling and Scientific Visualization	Effectively design and visualize predictive models in research	Machine Learning (Scikit-learn), Deep Learning (Keras), visualization (Plotly, Dash)	Challenges in modeling complex data in a scientific context	4h	Formative
C24, C25	**Lecture 7: Introduction to Deep Learning and Advanced Modeling**	Understanding simple neural networks to model complex phenomena	Deep Learning with Keras (Basic MLP), Overfitting, Layer Visualization	Why and how to use neural networks in applied research?	4h	Formative
Total Lecture Hours					**20H**	Formative

3.8. Program of Proposed Workshops

The practical workshops are evaluated summatively. They aim to concretely validate the skills acquired by doctoral students through the direct application of AI tools and techniques in real-world scenarios.

Detailed Table of Practical Workshops (28 Practical Hours)

Targeted Skills	Practical Workshops	Workshop Objectives	Proposed Activities	Scientific Issues Addressed	Duration	Evaluation
C26, C13, C21	**Workshop 1:** Hands-on Experience with Open Source Tools	Master Zotero, Jupyter, Overleaf for efficient scientific document management	Installation and practical use of tools	Challenges in managing and reproducing scientific work	3h	Summative
C21, C13	**Workshop 2:** Open Collaborative Work on Scientific Projects	Master Git/GitHub and Overleaf for collaborative project management	Collaborative practical work, version control	Complexity of managing collaborative scientific projects	3h	Summative
C19, C23	**Workshop 3:** Data Cleaning and Preprocessing in Scientific Research	Efficiently prepare experimental data for analysis	Cleaning and processing with Pandas, OpenRefine	Challenges in preparing experimental data	2h	Summative
C20, C23	**Workshop 4:** Applied Statistical Analysis on Real Data	Conduct concrete statistical analyses on scientific data	ANOVA, correlations, applied statistical tests	Complexity of statistical analyses in research	4h	Summative
C22, C24	**Workshop 5:** Practical Implementation of Predictive Models (ML & DL)	Build and validate predictive models applied to concrete problems	Implementation with Scikit-learn and Keras	Classification and prediction of scientific results	4h	Summative
C25, C24	**Workshop 6:** Designing Interactive Dashboards	Develop dynamic scientific visualizations for research	Dashboard creation with Dash, Streamlit	Challenges in clearly presenting complex scientific results	4h	Summative
C24, C25	**Workshop 7:** Design of interactive dashboards	Developing dynamic scientific visualizations for research	Creating dashboards with Dash, Streamlit	Difficulty in clearly presenting complex scientific results	4h	Summative
C19, C20, C21, C22, C23, C24	**Workshop 8:** Complete Experimental Scientific Project with AI	Conduct a complete project involving analysis, modeling, and data interpretation	Full application (preprocessing, analysis, modeling, validation) on real datasets	Challenges in extracting and leveraging insights from large scientific databases	4h	Summative
Total Workshop Hours					**28H**	Summative

4. DETAILED PROGRAM

Axis 1: Role and Use of Open Source Tools in AI (8 hours)

The use of open-source tools is essential to ensure reproducible, collaborative, and accessible research, particularly in fields that leverage artificial intelligence technologies. This research area allows doctoral students to familiarize themselves with these essential tools and develop autonomy in their use.

Theme 1: Discover the ethical and legal principles of open-source licenses (GPL, MIT, Apache).
Theme 2: Overview of open source tools in research and teaching.
Theme 3: Exploiting open source tools for management and collaboration.

1- Discover the Ethical and Legal Principles of Open Source Licenses (GPL, MIT, Apache) (2h)

- **Objective:** Understand the legal and ethical principles of open source licenses, as well as their implications for scientific research and software development.

- **Tools:** No specific tools required, but analysis of license documents and open source projects (e.g., GitHub).

- **Content:**

- *Introduction to the concept of free and open source software*
- *Differences between free and proprietary licenses*
- *Presentation of the main open source licenses:*
    - **GPL**: copyleft, reciprocity
    - **MIT**: permissive, simple
    - **Apache 2.0**: permissive, includes patent clause
- *Comparison of licenses (summary table)*
- *Legal responsibilities and best practices (citations, compliance with clauses)*
- *Ethical issues related to the free dissemination of knowledge*

- Practical Workshop:
    - **Case Study:** Identify the license of an open source project on GitHub
    - **Discussion:** What type of license should be chosen for an academic project?

2- Overview of open source tools in research and teaching (2 hours)

- **Objective:** Discover the main open-source tools and understand their benefits for research and education.
- **Tools:** *Linux, LibreOffice, Zotero, R, Python, Jupyter Notebook*
- **Content:**

- *Introduction to Free and Open Source Software*
  - History and principles of open source.
  - Difference between open source and proprietary software.
  - Advantages of open source for scientific research (transparency, reproducibility, collaboration).
- *Presentation of open source tools according to their use (office automation, bibliographic management, data analysis).*
- *Comparison with proprietary tools.*
- *Ethical and legal issues*
  - Open source licenses (GPL, MIT, Apache, etc.).
  - Responsibilities and good usage practices.
- *The role of open source in the use, promotion, and development of artificial intelligence.*

- **Practical workshop:**

Test different tools based on doctoral students' needs (e.g., organizing a bibliography with Zotero, and running a Python script on Jupyter).

3- Using open source tools for management and collaboration (4 hours)

- **Objective:** Use open-source tools to facilitate project management and collaborative work in research.
- **Tools:** *Git/GitHub, Nextcloud, Overleaf, Jitsi Meet*
- **Content :**

- *Presentation of management and collaboration tools.*
- *Version control with Git and collaboration via GitHub.*
- *Sharing and collaborative editing of scientific documents.*

- **Practical workshop:**

Work as a team on a research project using GitHub and Overleaf for collaborative writing.

Axis 2: Learning and communication augmented by AI (8 hours)

The second axis of the training program aims to explore how AI can enhance learning and improve scientific communication. It is structured around three complementary themes, each offering a specific approach to the use of AI tools in an academic context.

Theme 1: Understanding the fundamental concepts of AI and its impact on education and communication.
Theme 2: Harnessing AI to improve writing, research, and critical analysis.
Theme 3: Developing a critical approach to AI tools for effective and ethical use.

Recommendation: Within each theme, the suggested tools offered offer free versions with sometimes limited functionality. It's advisable to explore these versions for an initial hands-on experience before considering subscriptions for advanced analytics. Furthermore, open-source alternatives may be preferred for greater flexibility.

1- Introduction to AI and its implications in learning and communication (2 hours)

- **Objective:** This theme aims to understand the foundations of AI, its evolution, and its impacts on learning and communication.
- **Tools:** *ChatGPT, Claude, Gemini, Copilot*
- **Content :**

- *Definition and history of AI*
  - Origin and evolution of artificial intelligence systems.
  - Difference between Symbolic AI, Machine Learning, and Neural Networks.
  - Impact of recent advances on education and communication.

- *Areas of application*
  - Education: smart tutorials, auto-correction, writing assistance.
  - Communication: synthesis and reformulation of texts, automatic translation.
  - Content production: article generation, automatic summarization.

- *Ethical issues and challenges*
  - Algorithmic Bias: How Does Data Influence Outcomes?
  - Transparency and explainability: why is it important to understand how AI works?
  - Data protection and regulation: best practices and legal framework.

- **Practical workshop:**

This workshop will explore the capabilities of AI for the synthesis and reformulation of academic texts while developing a critical perspective on the productions generated.

- Getting started with the tools: using ChatGPT, Claude, Gemini, and Copilot.

- Experimentation: generation of summaries and reformulation of academic passages.
- Critical analysis: comparison of results, identification of limitations and biases.
- Practical application: application on a document chosen by the doctoral students.
- Collective feedback and formative assessment: discussions on learning and adjustments.

2- Assistance with scientific writing with AI (3 hours)

- **Objective:** This theme aims to improve the editorial quality of academic work by exploiting AI tools for correction, reformulation, and linguistic adaptation.
- **Tools:** *Grammarly, LanguageTool, ChatGPT, DeepL*
- **Content :**

- *Correction and optimization of text*
  - Identification of linguistic errors (grammar, spelling, punctuation).
  - Suggestions for improving the structure and readability of an academic text.
- *Improving style and adapting editorial tone*
  - Reformulation and enrichment of scientific vocabulary.
  - Adjustment of the language level according to the target audience and academic requirements.
- *Multilingual translation and harmonization*
  - Using AI tools to translate and adapt articles into multiple languages.
  - Comparison between different AI solutions and their impacts on translation fidelity

- **Practical workshop:**

This workshop will consist of putting into practice the skills acquired by using AI tools to improve the writing quality of a scientific text.

- Automatic analysis and correction of a text using Grammarly and LanguageTool.
- Reformulation and improvement of editorial style with ChatGPT.
- Translation and adaptation of an academic passage into several languages using DeepL and open-source alternatives.
- Comparison of results obtained between free and paid tools.

3- Documentary research and critical analysis assisted by AI (3 hours)

- **Objective:** This theme aims to harness the capabilities of AI to optimize bibliographic research and improve the critical evaluation of academic sources. It allows doctoral students to more efficiently access relevant publications while developing a critical perspective on the reliability of information.
- **Tools :** *Elicit , Semantic Scholar, Scite , Zotero with AI*
- **Content :**

- *Optimization of document research*
  - Using AI-powered scientific search engines to identify relevant articles.
  - Exploration of academic databases and automated extraction of key information.

- *Synthesis and summary of publications*
  - Intelligent summary generation for rapid analysis of academic content.
  - Comparison of automated and manual approaches in source evaluation.
- *Critical evaluation and validation of references*
  - Analysis of citations and relationships between publications to judge their impact.
  - Considering algorithmic biases and the limitations of AI tools in scientific research.

- **Practical workshop :**

The workshop will consist of comparing a classic documentary search with an AI-assisted search.

- Researching an academic topic using **Google Scholar** and a traditional approach.
- A second research with **Elicit, Semantic Scholar and Scite**, to measure the effectiveness and limitations of AI in document analysis.
- A discussion on algorithmic bias and the ethics of using these tools.

Axis 3: Research and analysis of data driven by AI (24 hours)

Axis III focuses on data acquisition, processing, and analysis using artificial intelligence (AI) tools. The objective is to master the fundamental steps of modeling and interpreting results through modern mathematical and algorithmic approaches.

1- Introduction to Data Science with Python (2 hours)

- **Objective:** Establish the technical foundations of data science with Python. Explore fundamental libraries, understand data structures, and master the basic exploration and inspection commands.
- **Tools:** *Jupyter Notebook, Google Colab, Python 3.8+, Pandas, NumPy*
- **Content :**

▪ *Data loading (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 ' )
```

▪ *DataFrame Overview*

```
df.head ()  # first 5 lines
df.tail ()     # last 5 lines
df .sample (3)          # 3 random lines
df . shape              # dimensions (rows, columns)
df . columns            # column names
df . dtypes             # types of each column
df .info( )   # full summary
```

▪ *Basic Statistics*

```
df . describe ()                    # numerical statistical summary
df [ 'column' ]. value_counts ()              # frequency of values
df [ 'column' ]. nunique ()         # number of unique values
```

▪ *Display configuration*

```
pd.set_option ( ' display.max_columns ' , None ) # display all columns
pd.set_option ( ' display.precision ' , 2 ) # number of decimal places
```

▪ *Column/Row Selection*

```
df [ 'brand' ]            # single column
df [ [ 'brand' , 'price' ]]           # multiple columns
df . iloc [0]             # first line
df . iloc [0:5]           # lines 0 to 4
df . loc [ df [ 'price' ] > 10000 ]     # conditional filtering
```

▪ *Creating and Deleting Columns*

```
df [ ' price_per_kg ' ] = df [ 'price' ] / df [ 'weight' ]
```

```
df . drop ( ' old_column ' , axis= 1 , inplace = True )
```

- *Sorting values*

```
df . sort _values ( by = 'price' , ascending = False )
```

- **Workshop :**
  - Upload a CSV file containing vehicle information
  - Show columns, types, and dimensions
  - Count unique values in a column (eg fuel)
  - Create a calculated column (`price_per_kg`)
  - Sort cars by descending price and extract the first 10

2- Data preprocessing and cleaning (3 hours)

- **Objective:** Clean, correct, and transform data to guarantee the quality, consistency, and robustness of analyses and predictive models.
- **Tools :** *Pandas, NumPy, Scikit-learn* (preprocessing), OpenRefine (optional)
- **Detailed content:**

- *Missing Value Detection*
  - Localization of `NaNs` by column or row

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

- *Handling missing values*

- Filling (imputation) by mean, median, constant, or advanced method:

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

- Imputation with SimpleImputer (Scikit-learn)

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

- *Outlier detection*

- Interquartile range (IQR) method

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

- Filtering by z-score

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

- *Removing duplicates*

- Search and eliminate repeated lines

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

```
df . drop _duplicates ( inplace = True )
```

- *String cleaning*

```
df [ ' city ' ] = df [ ' city ' ]. str.strip () . str.lower () . str .replace ('-', ' ')
df [ ' city ' ] = df [ ' city ' ]. str.normalize ( 'NFKD ' )
```

- *Standardization of data types*

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

- *Creating Derived Columns*

- Generate new useful variables from existing ones:

```
df [ ' price_per_kg ' ] = df [ 'price' ] / df [ 'weight' ]
df [ ' age ' ] = 2024 - df [ ' year ' ]
```

- *Encoding of categorical variables*

- One-hot encoding and label encoding

```
p.d. get_ dummies ( df , columns=['fuel'])
from sklearn.preprocessing import LabelEncoder
the = LabelEncoder ( )
df [' gender_code '] = le.fit_transform ( df [ ' gender ' ] )
```

- *Normalization and standardization*

- Scaling for sensitive models (KNN, linear regression, etc.)

```
from sklearn.preprocessing import MinMaxScaler , StandardScaler
MinMaxScaler ( ) . fit _transform ( df [[ ' revenue ' ]])
StandardScaler ( ) . fit _transform ( df [[ ' revenue ' ]])
```

- *Detecting business inconsistencies*

- Example: a weight less than 300 kg or an age greater than 120 years

```
df [ df [ ' weight ' ] < 300 ]
df [ df [ 'age' ] > 120 ]
```

- **Practical workshop:**
  - Identify and address missing, outlier, and inconsistent values
  - Clean up text columns, unify formats
  - Create useful columns for modeling
  - Apply normalization and encode categories

3- Grouping, joins, and aggregations (2 hours)

- **Objective:** Organize and summarize data according to several axes: type of vehicle, fuel, period, etc. Obtain KPIs, create cross-tabulations, and cross-reference several sources.
- **Tools:** Pandas
- **Detailed content:**

- *Averages, sums, counts by category*

```
df . groupby ( 'brand' )[ df ]. mean()
df.groupby ( 'fuel' ) [ 'power' ] . sum ()
df . groupby ( 'type' ). size()                    # Number of elements per type
```

- *Multiple aggregations with agg ( )*

```
df . groupby ( 'brand' ). agg ({
  'price' : [ 'mean' , 'max' , 'std' ],
  ' weight ' : 'median'
})
```

- *Aggregation across multiple columns simultaneously*

```
gb = df . groupby ([ 'brand' , 'fuel' ])
gb [ 'price' ]. mean ()
```

- *Pivot Tables*

```
pd . pivot _table ( df , values = 'price' , index = 'brand' , columns = 'fuel' , aggfunc = 'mean' )
```

- *Sorting aggregated groups*

```
df . groupby ( 'brand' )[ 'price' ]. mean() . sort_values ( ascending = False )
```

- *Merges between DataFrames (relational joins)*

```
p.d. merge ( df_clients , df_commands , on= ' client_id ' , how= 'inner' )
p.d. merge ( df1 , df2 , left_on = 'id' , right_on = ' product_id ' , how= 'outer ')
```

- *Concatenation (stacking multiple arrays)*

```
pd . concat ( [ df1 , df2 ], axis = 0 )                    # line by line
pd . concat ( [ df1 , df2 ], axis = 1 )                    # column by column
```

- **Workshop:**

  - Create a cross-tabulation of average prices by brand and fuel type
  - Merge customer and order data to calculate total spend per customer

4- Exploratory analysis & statistics (2 hours)

- **Objective:** Acquire the statistical bases to better understand the variables of a data set: central tendencies, dispersion, variance, simple correlation, and clear visualization.
- **Tools :** Pandas, NumPy, Seaborn, Matplotlib, Statsmodels
- **Content simplified :**

- *Basic Statistics*

- Mean, median, standard deviation, quartiles

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

- *Variance and covariance*

```
df . var ( )
df . cov ( )
df . corr ()                              # Pearson correlation
```

- *Visualization of distributions*

- Histogram and box plot

```
import seaborn ace sns
sns . histplot ( df [ 'price' ], bins = 20 , kde = True )
```

```
sns . boxplot (x= df [ 'price' ])
```

- *Analysis of variance between groups (simplified ANOVA)*

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

- *Simple Linear Regression with Statsmodels*

```
import statsmodels.api ace sm
X = df [[ ' weight ' ]]
y = df [ 'price' ]
X = sm . add_constant ( X )                    # Adding the intercept t
model = sm . OLS ( y , X ). fit()
print ( model.summary ( ) )
```

- Workshop:
  - Visually compare price distributions between brands
  - Display the statistical values of a numeric variable
  - Check standard deviations and quartiles between categories

5- Data Visualization (3 hours)

- **Objective:** To visually represent relationships between variables to facilitate interpretation.
- **Tools:** *Matplotlib, Seaborn, Plotly*

- *Histogram:*

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

- *Point cloud:*

```
sns . scatterplot ( x = ' weight ' , y = ' price ' , hue = ' brand ' , data = df )
```

- *Correlation heatmap :*

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

- *Interactive visualizations with Plotly :*

```
import plotly.express ace px
px . scatter ( df , x = ' weight ' , y = ' price ' , color = ' fuel ' )
```

- Workshop:
  - Visual comparison of car models based on their performance and price.

6- Predictive Modeling & Deep Learning (10 hours)

- **Objective :** To build and compare prediction models (classic and simple in deep learning).
- **Tools :** *Scikit-learn, XGBoost , Keras , SHAP, Joblib , pycaret*

- **Module structure (10 hours):**
  - ***1h**: Preparation and separation of the data set*
  - ***2h**: Linear and regularized regression (Lasso, Ridge)*
  - ***2h**: Classification (LogisticRegression , SVM, KNN)*
  - ***2h**: Advanced methods (RandomForest, XGBoost, LGBM)*
  - ***1h**: Introduction to Deep Learning (Keras + dense network)*
  - ***2h**: Advanced evaluation, interpretability (SHAP), backup*

- **Detailed content:**

▪ *1) Preparation of the 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) Regression models:*

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

▪ *3) Random forest regression:*

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

▪ *4) Regression by boosting :*

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

▪ *5) Classification models:*

```
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) Simple neural networks with 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) Evaluation and interpretability:*

```
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) Explainability with SHAP:*

```
import shape
explainer = shap . Explainer ( model.predict , X_test )
shap_values = explainer ( X_test )
```

```
shap . plots . beeswarm ( shap_values )
```

- ***9) Using AutoML with PyCaret :***
  - ***Pyacaret*** is an open-source Python library that makes it easy to develop machine-learning models. It's a simplified alternative to scikit-learn, designed to accelerate and automate the entire modeling lifecycle: from data preparation to deployment.
  - *Very simple interface: just a few lines of code are enough.*
  - *Automatic data preprocessing (missing values, encoding, normalization, etc.).*
  - *Automatic comparison of multiple algorithms.*
  - *Supports many use cases:*
    - *Classification*
    - *Regression*
    - *Clustering*
    - *Anomaly detection*
    - *Text analysis (NLP)*
    - *Time series (forecasting)*

```
# Example: Automatic regression on a vehicle dataset

from pycaret.regression import *
import pandas ace pd

# Loading the dataset
df = pd . read_csv (" cars.csv" )                # contains weight, power, price, etc.

# Initializing the AutoML environment
reg = setup ( data = df , target = 'price' , session_id = 123, verbose = True )

# Automatic comparison of all models
best _model = compare_models ()

# Visualization of residuals and variable importance
plot_model ( best_model , plot = 'residuals ' )

# Visualization of variable importance
plot _model ( best_model , plot = ' feature ' )

# Saving the model
save_model ( best_model , ' auto_price_model ' )
```

- Workshop:
  - Model comparison: linear regression vs random forest vs neural network.
  - **Optional**: Rework some Machine Learning models with ***PyCaret***.

7- Creation of an interactive dashboard (2 hours)

- Objective**:** Create a simple interactive application using only Python and standard libraries to visualize and analyze data. Integrate **TensorBoard** for displaying and tracking machine learning model metrics.
- Tools**:** *Pandas, Matplotlib, ipywidgets , TensorBoard (via TensorFlow)*
- Detailed example:

- **Interactive visualization of car prices by weight**

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

 # Loading a sample dataset

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

 # Interactive function to visualize data

```
def view_max_price ( max_weight ):
   df _filter = df_cars [ df_cars [ 'weight' ] <= max_weight ]
   plt . figure ( figsize =( 8,4 ))
   plt.scatter ( df_filter [ ' weight ' ] , df_filter [ 'price' ], color = ' blue' )
   plt . title ( f'Price of cars up to { max_weight } kg')
   plt . xlabel (' Weight (kg)')
   plt . ylabel ('Price (€)')
   plt . grid ( True )
   plt.show ( )
```

 # Interactive widget

```
interact ( visualize_max_price , max_weight = IntSlider ( min = 800 , max = 2000 , step = 100 , value = 1500 ))
```

TensorBoard integration to visualize the training of a simple model

```
import tensorflow as tf
import numpy as np

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

# Creating a simple model
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 ')

# Configuring TensorBoard
log _dir = "logs/fit"
tensorboard_callback = tf . keras.callbacks.TensorBoard ( log_dir = log_dir , histogram_freq = 1 )

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

 # Visualization with TensorBoard (in a Jupyter or Colab notebook):

```
% load _ext tensorboard
% tensorboard -- logdir logs/fit
```

- ## Workshop:
  - #1. Download and upload a CSV file containing real vehicle information.
  - #2. Display descriptive statistics of the loaded data.
  - #3. Adapt the previous interactive function to display prices according to another criterion, such as the horsepower or year of the vehicles.
  - #4. Experiment with different interactive widgets to enrich the user experience.
  - #5. **Optional**: Train a simple model and display the results with **TensorBoard**

Axis 4: Use of AI to solve research problems (8 hours)

This theme aims to help doctoral students apply artificial intelligence through case studies adapted to different scientific fields. They will learn to master the development pipeline of an AI process, from data preprocessing to analysis and results generation, using specialized tools. The emphasis will be placed on experimentation with real-world datasets and on the creation of interactive dashboards to visualize and interpret the results in a clear and relevant way. These lessons will be put into practice through dedicated workshops, allowing doctoral students to gain hands-on experience with the techniques covered.

1- Creation of interactive dashboards for data analysis (2 hours)

- **Objective:** Learning how to create interactive and dynamic dashboards is essential for effectively analyzing and visualizing scientific data. These tools enable in-depth data exploration and facilitate clear communication of results, making complex information more accessible and understandable.
- **Tools:** ***Power BI*** *(or* ***Tableau)****, Shiny (R)*
- **Content :**

- *Introduction to Dashboards*
  - Presentation of the tools (Power BI, Tableau, Shiny).
  - Basics of data visualization.
  - Structuring a dashboard (objectives, key indicators, and impact on research).
- *Creating a dashboard*
  - Data import and preparation.
  - Creation of simple visualizations (histograms, trend lines).
  - Integration of interactive features (filters, drill-down).
- *Exploratory analysis*
  - Explore data interactively.
  - Identify trends or anomalies.

- **Practical workshop:**

  - Doctoral students create a simple dashboard from a scientific dataset (e.g. environmental or medical data).
  - They explore the data and identify trends or anomalies.

2- Case studies: application of AI to various fields (3 hours)

- **Objective:** To use artificial intelligence techniques to address real-world problems in various fields. By applying AI to specific challenges, we demonstrate its usefulness and potential to transform research, offering innovative solutions and opening new perspectives for solving complex problems.

- **Tools:** *depending on the field (SHS or ST) and the project chosen.*
- **Content :**

- *Presentation of AI use cases in various fields.*
- *Explanation of AI techniques adapted to the field of study.*
- *Implementation of an AI model adapted to the chosen problem.*
- *Analysis of results and proposal for improvements.*
- *Discussion on the specific issues and challenges of this application.*
- *Critical study: strengths, weaknesses, and limitations of AI in academic research.*

Project examples for Humanities and Social Sciences (HSS)

- Objective: Digitize and analyze old documents (e.g. letters, newspapers) to extract key information (dates, places, people, etc.).
- Techniques: OCR (optical character recognition), NER, text classification.
- Tools: Tesseract (OCR), SpaCy, Python.

Example of a project for Science and Technology (ST)

- Objective: Classify medical images (e.g., x-rays, MRIs) to detect pathologies.
- Techniques: Convolutional neural networks (CNN), transfer learning.
- Tools: Keras, TensorFlow, OpenCV.

Examples of interdisciplinary projects

- Objective: To develop a chatbot capable of answering questions in both SHS and ST (e.g. questions on the history of science and technical concepts).
- Techniques: Language models (GPT, BERT), natural language processing (NLP).
- Tools: Rasa, Dialogflow, Transformers.

- **Practical workshop:**

- **Group work on a case study**: Doctoral students will apply AI techniques to a real-life problem.
- **Model Implementation**: PhD students will be required to train and evaluate an AI model.
- **Analysis of results**: Doctoral students will have to interpret the results and suggest improvements.

3- Experimentation on scientific databases (3 hours)

- **Objective:** To provide doctoral students with hands-on experience working with real scientific data sets. They will explore, preprocess, and analyze these data while addressing the challenges specific to each field. Through this approach, they will learn to manipulate diverse datasets, extract relevant information, and develop expertise in data management and analysis. This sub-axis will allow them to become familiar with complex data and better understand its use in a research context.
- **Tools:** *Python, Pandas, NumPy, Matplotlib/Seaborn, Scikit-learn, Jupyter Notebook*
- **Content :**

- *Introduction to Scientific Datasets*
  - dataset types (tabular data, time series, images, texts, etc.).

  - Specific challenges related to scientific datasets (data quality, volume, heterogeneity, bias).
  - Examples of public scientific datasets (eg: UCI Machine Learning Repository, Kaggle, Open Data Portals).
- *Data exploration and preprocessing*
  - Data cleaning techniques (missing value management, normalization, standardization).
  - Data exploration (descriptive statistics, visualization of distributions).
  - Data preparation for analysis (feature engineering, variable selection).
- *Data analysis*
  - Application of statistical analysis techniques (correlations, hypothesis tests).
  - Use of simple Machine Learning models for data exploration (e.g. clustering, linear regression).
  - Visualization of results (graphs, maps, etc.).
- *Critical study*
  - Discussion on the limitations of scientific datasets (data quality, potential biases, representativeness).
  - Reflection on the impact of methodological choices on research results.

5. BIBLIOGRAPHICAL REFERENCES

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.