Course Catalog
Master for Smart Data Science

ACADEMIC YEAR 2026 / 2027

ENSAI
École nationale
de la statistique
et de l'analyse
de l'information

List of courses

General Presentation and Objectives

The world is producing previously unimaginable amounts of data every second. This data could help to understand and improve our society, to predict and prevent, to combat diseases and generally improve life. Extracting valuable information and creating knowledge from the massive and heterogeneous data require skills in statistical modelling, machine learning algorithms, as well as computer science. The synergy of these academic fields, oriented towards their application, is the guiding idea of the Master for Smart Data Science at ENSAI.

ENSAI is part of the network of prestigious higher-education establishments in France known as Grandes Ecoles, or specialized graduate schools. ENSAI trains its students to become qualified, high-level specialists in information processing and analysis.

The graduates of this Master will be capable of creating and implementing methodologies and algorithms for analyzing large flows of data arriving from different sources, of using statistical tools and machine learning algorithms to identify correlations, effects, patterns and trends in data, and of formalizing predictions. As such, they will be qualified for data scientist and artificial intelligence jobs in industry, marketing, banking and insurance, media, or further pursuing a PhD.

This Master's program is composed of 1 semester of coursework at ENSAI, followed by a four to six-month paid internship in France or abroad within the professional world, academia, or research laboratories.

Since this program welcomes students with varying academic levels and skills in Computer Science, Applied Mathematics and Statistics, preliminary coursework is put in place to bring all students to the same scientific level in these fields, with respect to their existing training, knowledge, and skills.

Curriculum – Program Overview and Credits

	Semester Hours	CTS Credits	Total in the block
UE–MSD01 – Machine Learning			
Machine Learning for Data Science	24	2.5	
Deep Learning	24	2.5	7
Dimension Reduction & Self-supervised Representation Learning	18	2	
UE–MSD02 – Models for Dependent Data			
Machine Learning for Time Series	18	2	5
High-Dimensional Time Series	24	3	
UE–MSD03 – Statistics for New Data			
Functional Data Analysis	24	3	5
Machine Learning for Natural Language Processing	18	2	
UE–MSD04 – Advanced Tools for Data Analysis & Computing			
Data Visualization	15	1	3
Parallel Computing & Data Science with R, Julia and Python	18	2	
UE–MSD05 – IT Tools			
IT Tools 1 (Hadoop & Cloud Computing)	18	2	5
IT Tools 2 (NoSQL, Big Data Processing with Spark)	24	3	
UE–MSD06 – Case Studies and Project			
Smart Data Project / or Research Project	24	3.5	5
Topics & Case Studies, Conferences / or Research Project	36	1.5	
TOTAL Semester 1	285 H	30 credits	
UE-MSD07- Internship End-of-Studies Internship	(4 to 6 months)		30
TOTAL Semester 2		30 credits	
TOTAL Academic Year	285 H	60 credits	

Prior to the start of the first semester, the students attend mandatory courses designed to reinforce different topics in Computer Science, Statistics, and Mathematics. The tentative list of these courses for September 2026 is the following:

GNU Linux & Shell Scripting	12 hrs
Reproducible Workflows	09 hrs
SQL	06 hrs
Statistical Language: R	09 hrs
Statistical Language: Python	12 hrs
Topics in Statistics	12 hrs
Topics in Probability & Analysis	09 hrs

List of Professors and Lecturers

Code	*Topic*	*Professor/Lecturer*
Preliminary 1	GNU Linux & Shell Scripting	Guillaume GRABE
Preliminary 2	Reproducible Workflows	Aymeric STAMM
Preliminary 3	SQL	Nikolaos PARLAVANTZAS
Preliminary 4	Statistical Language: R	Aymeric STAMM
Preliminary 5	Statistical Language: Python	Sébastien HERBRETEAU
Preliminary 6	Topics in Statistics	Marie-Pierre ETIENNE
Preliminary 7	Topics in Probability & Analysis	François PORTIER
MSD 01-1	Machine Learning for Data Science	François PORTIER Thibault PAUTREL
MSD 01-2	Deep Learning	Johann FAOUZI
MSD 01-3	Dimension Reduction & Self-supervised Representation Learning	Titouan VAYER
MSD 02-1	Machine Learning for Time Series	Simon MALINOWSKI
MSD 02-2	High-dimensional Time Series	Lionel TRUQUET
MSD 03-1	Functional Data Analysis	Valentin PATILEA
MSD 03-2	Machine Learning for Natural Language Processing (NLP)	Guillaume GRAVIER
MSD 04-1	Data Visualization	Etienne MADINIER
MSD 04-2	Parallel Computing & Data Science with R, Julia and Python	Aymeric STAMM Emmanuel PILLIAT
MSD 05-1	IT Tools 1 (Hadoop & Cloud Computing)	Shadi IBRAHIM
MSD 05-2	IT Tools 2 (NoSQL, Big Data Processing with Spark)	Nikolaos PARLAVANTZAS Hervé MIGNOT
MSD 06-1	Smart Data Project	Industrial/lab partners
MSD 06-2	Topics, Case Studies, Conferences	
	Bandit Theory	Romaric GAUDEL
	Some Recent Advances for Big Data Processing in the Cloud	Shadi IBRAHIM
	Stochastic Optimization Methods for Machine Learning	Rémi LELUC
	Case Studies in Smart Data	Thomas ZAMOJSKI
	Power BI	Franck ORAGA
MSD 07-1	End-of-Studies Internship	

Preliminary Courses

Preliminary 1 – MSD - Before the start of the 1st Semester

GNU Linux & Shell Scripting

Professor	:	Guillaume GRABE (Airnity)
ECTS Credits	:	0 (preliminary course)
Lectures and Tutorials	:	12 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	0
Teaching language	:	English
Software & Packages	:	Linux + Shell (already installed on ENSAI's computer)
Teaching materials	:	A computer (lent by ENSAI) + internet connection

Learning Outcomes

This class teaches students the concepts that they should understand before they start working with GNU/Linux. During this course, students will configure a distribution on their computer and learn how to interact with the shell, from basic tasks (navigation, file edition, network configuration) to more advanced operations with shell scripting. GNU/Linux is essential in particular when using and developing Big Data technologies.

Prerequisites

A computer (lent by ENSAI) + internet connection

Subjects Covered

1. GNU/Linux
- Introduction to GNU/Linux
- Installing a distribution
- The shell
- Users, groups, permissions
- Packages management
- Network management

2. Shell scripting
- Shell scripting principles
- Variables in the shell, operations on variables
- Conditional expressions, basic statements, functions
- Regular expressions

Bibliography

1. https://wiki-dev.bash-hackers.org/
2. http://tldp.org/index.html
3. B. FOX and C. RAMAY, Bash Reference Manual, Free Software Foundation

Preliminary 2 – MSD - Before the start of the 1st Semester

Reproducible Workflows

Professor	:	Aymeric STAMM (CNRS & ENSAI)
ECTS Credits	:	0 (preliminary course)
Lectures and Tutorials		9 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	0
Teaching language	:	English
Software & Packages	:	
Teaching materials	:	

New course – Its description will be added soon.

Preliminary 3 – MSD - Before the start of the 1st Semester

SQL

Professor	:	Nikolaos PARLAVANTZAS (IRISA Rennes)
ECTS Credits	:	0 (preliminary course)
Lectures and Tutorials		6 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	0
Teaching language	:	English
Software & Packages	:	PostgreSQL
Teaching materials	:	Slides and lab subjects on Moodle

Learning Outcomes

Understand the fundamentals of relational databases and the SQL language.
Learn how to use SQL to retrieve, analyze, and manipulate data.
Gain hands-on experience in applying SQL using a popular relational database management system.

Prerequisites

Basic computer literacy

Subjects Covered

- Introduction to relational databases and SQL (querying and manipulating data, designing databases)
- Practical exercises using PostgreSQL

Bibliography

Many online resources are available

Preliminary 4 – MSD - Before the start of the 1st Semester

Statistical Language: R

Professor	:	Aymeric STAMM (CNRS)
ECTS Credits	:	0 (preliminary course)
Lectures and Tutorials	:	9 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	9 hrs
Teaching language	:	English
Software & Packages	:	R
Teaching materials	:	Slides and tutorials on Moodle

Learning Outcomes

Students should be able to:
1. Understand and navigate the R ecosystem at an elementary level.
2. Perform simple data wrangling and basic statistical analyses using the tidyverse ecosystem of R packages.

Prerequisites

A laptop with R and RStudio installed. No prior programming experience is required.

Subjects Covered

This course is recommended for those without prior programming experience in R, and wish to acquire a basic foundational knowledge on how to use R for data analysis using the tidyverse framework. Some topics covered include basic data structures, data transformation and visualizations and simple statistical analyses. Students will be introduced to a few useful packages in R.

Bibliography

1. HADLEY WICKHAM & GARRETT GROLEMUND, R for Data Science, O'Reilly, 2017: https://r4ds.hadley.nz/
2. GROLEMUND G., Hands-On Programming with R, O'Reilly, 2014. https://rstudio-education.github.io/hopr/
3. https://ggplot2-book.org/

Preliminary 5 – MSD - Before the start of the 1st Semester

Statistical Language: Python

Professor	:	Sébastien HERBRETEAU (ENSAI)
ECTS Credits	:	0 (preliminary course)
Lectures and Tutorials	:	12 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	4 hrs
Teaching language	:	English
Software & Packages	:	Python, NumPy, PyTorch, Matplotlib, Pandas
Teaching materials	:	Moodle

Learning Outcomes

Python is a programming language used for many different applications. In this practical course, students will start from the very beginning, with basic arithmetic and variables, and learn how to handle data structures, such as Python lists or dictionaries, NumPy arrays and PyTorch tensors. Students will learn about Python functions and classes, control flow and data visualizations with Matplotlib. At the end of the lecture, the students are expected to know how to code with Python and how to use this language for data science and deep learning.

Prerequisites

Experience in programming with possibly other languages

Subjects Covered

- Setting up your Python environment
- Write functions using control flow tools and manage files input and output
- Introduction to object orienting programming.
- Jupyter Notebook
- NumPy
- Matplotlib
- PyTorch

Bibliography

1. Python documentation http://docs.python.org/
2. VANDERPLAS J., Python Data Science Handbook: Essential Tools for Working with Data, O'Reilly
3. LUTZ M., ASCHER D., Learning Python, O'Reilly
4. LANGTANGEN H.P, Python Scripting for Computational Science, Springer
5. STEPHENSON B., The Python Workbook, Springer
6. How to Think Like a Computer Scientist: Interactive Edition https://runestone.academy/ns/books/published/thinkcspy/index.html

Preliminary 6 – MSD - Before the start of the 1st Semester

Topics in Statistics

Professor	:	Marie-Pierre ETIENNE (ENSAI)
ECTS Credits	:	0 (preliminary course)
Lectures and Tutorials	:	12 hrs
Estimated personal workload (beyond lecture and tutorial time)	:	12 hrs
Teaching language	:	English
Software & Packages	:	R and Python
Teaching materials	:	Slides on Moodle and scripts on R

Learning Outcomes

This course uses the introduction to Principal Component Analysis and Linear regression to cover the concepts of Linear Algebra useful in Data science. The concepts will be illustrated by applications using R (alternative options using Python will be provided but not discussed during the labs)
At the end of the lectures, the students are expected to be comfortable with basic Linear algebra concept and how to analyze multivariate data.

Prerequisites

Notions on vectors, matrices.

Subjects Covered

- Principal Components Analysis: Algebraic derivation of the principal components of a random vector, Geometric properties of the principal components as a least squares approximation. The Singular Value decomposition point of view. Rescaling principal components. Choosing the number of components. Interpreting the components. Representation of individuals and variables, Determination of the number of components. Contribution of variables and individuals to the PCA construction.
- LInear regression: Gaussian random vector, Maximum likelihood estimation, Geometric properties of the maximum likelihood estimation, difference with the PCA. Assessing the importance of the individual in the determination of the regression. Assessing the quality of the prediction (Mean Square Error). Choosing the number of variables.
- Probabilistic PCA: Marrying MLE and PCA.

Bibliography

1. EVERITT, B.S, An R and S-Plus companion to multivariate analysis, Springer, 2005.
2. EVERITT, B.S, Dunn, G. Applied multivariate data analysis. Hodder Education, 2001.
3. HUSSON, F., Le, S., PAGES, J. Exploratory multivariate analysis by example using R. CRC Press, 2011.
4. JOHNSON, R.A., WICHERN, D.W, Applied multivariate statistical analysis, Pearson Education, 2007
5. PETERSEN, Kaare BRANDT, and Michael Syskind PEDERSEN. "The matrix cookbook." Technical University of Denmark 7.15 (2008): 510.
6. BANERJEE Sudipto and ROY Anindya. Linear algebra and matrix analysis for statistics. Vol. 181. Boca Raton: Crc Press, 2014.

Preliminary 7 – MSD - Before the start of the 1st Semester

Topics in Probability and Analysis

Professor	:	François PORTIER (ENSAI)
ECTS Credits	:	0 (preliminary course)
Lectures and Tutorials	:	9 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	12 to 24 hrs depending on the student's knowledge
Teaching language	:	English
Software & Packages	:	N/A
Teaching materials	:	Blackboard

Learning Outcomes

This course serves as an introduction/refresher into many probability concepts that will be useful to model and predict the behavior of real data. At the end of the course, the students are expected to be able to:

- know how to model a dataset by a random process, and test the accuracy of their modelization
- simulate basic and slightly more involved random variables using R and/or Python
- understand the notions of convergence and distances between probability distributions
- know how to compute a gradient, a Hessian, or use the fact that a function is convex.

Prerequisites

Basic probability notions

Subjects Covered

- Basic refresher: random variables, expectation, variance, independence…
- Sampling random variables : basic Python and R functions, rejection and inverse transform sampling
- Convergence of random variables, law of large numbers and central limit theorem
- Comparison of probability distributions : distances, Q-Q plots.
- Notions of multivariate analysis: gradient, Hessian, convexity…

Bibliography

Books

1. GRIMMETT G.R. & STIRZAKER D.R., Probability and Random Processes, Oxford Sciences Publications, 1992 (2nd edition).
2. BLITZSTEIN, J. K., HWANG, J., Introduction to Probability, Second Edition. United States: CRC Press, 2019.

Online resources

1. Introduction to Probability, Statistics and Random Processes, H. Pishro-Nik, 2014
2. Seeing Theory: a Visual Introduction to Probability and Statistics

First Semester

1st Semester

TEACHING UNIT MSD-01: MACHINE LEARNING

ECTS Credits	: 7
Estimated personal workload (beyond lecture and tutorial time)	: 50 to 60 hrs
Lectures and Tutorials	: 66 hrs

Learning Objectives of the Teaching Unit

Introduce fundamental and modern machine learning approaches and provide computing tools for effective implementation. Topics in model/feature selection and regularization methods, regression trees, aggregation methods and support vector machine (more generally RKHS regression), as well as neural network methods and algorithm will be presented. Dimension reduction techniques are also presented. The students are expected to know the main up to date algorithms and to be able to implement them.

Description

The Machine Learning unit includes 3 courses:

1. Machine Learning for Data Science
2. Deep Learning
3. Dimension reduction & Matrix completion

Machine Learning for Data Science is a general and introductory course on Machine Learning. It will cover most of the techniques used in Machine Learning. Deep Learning is a more specific course on the use of Neural networks in Machine Learning. They have become one of the leading class of algorithms — due notably to their success in image processing. The last course is about Dimension reduction in Machine Learning. It covers several sets of techniques that are essential to treat large scale data.

Acquired Skills

Knowledge of a large panel of algorithms, use of modern machine learning approaches for complex data problems, implementation of algorithms using packages and notebooks.

Prerequisites

Regression models, Notions of probability theory, Linear algebra and geometry, Algorithm complexity.

UE-MSD01 – Machine Learning – MSD 01.1 - 1st Semester

Machine Learning for Data Science

Professors	:	François PORTIER (ENSAI) Thibault PAUTREL (L2S)
ECTS Credits	:	3.5
Lectures and Tutorials		30 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	5 to 10 hrs
Teaching language	:	English
Software	:	Python and R
Course materials	:	Slides and lecture notes

Learning Outcomes

Upon completing this course, students should be able to:

- select the appropriate methods;
- implement these statistical methods;
- compare leading procedures based on statistical arguments;
- assess the prediction performance of a learning algorithm;
- apply these key insights into class activities using statistical software.

Prerequisites

Linear algebra, probability, optimization

Subjects Covered

This course focuses on supervised learning methods for regression and classification. Starting from elementary algorithms such as ordinary least squares, we will cover regularization methods (crucial in large scale learning), nonparametric decision rules such as support vector machine, the nearest neighbor algorithm and CART.
Finally, bagging and boosting techniques will be discussed while presenting random forest and XGboost algorithm.

We shall focus on methodological and algorithmic aspects, while trying to give an idea of the underlying theoretical foundations. Practical sessions will give the opportunity to apply the methods on real data sets using either R or Python. The course will alternate between lectures and practical lab sessions.

Evaluation

Final exam and computer class

Bibliography

1. HASTIE T., TIBSHIRANI R., FRIEDMAN J.H., The elements of statistical learning: data mining, inference and prediction; 2009
2. JAMES G., WITTEN D., HASTIE T., & TIBSHIRANI R., An Introduction to Statistical Learning. New York: Springer. R; 2013.

UE-MSD01 – Machine Learning – MSD 01.2 - 1st Semester

Deep Learning

Professors	:	Johann FAOUZI (ENSAI)
ECTS Credits	:	2.5
Lectures and Tutorials	:	24 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	24 hrs
Teaching language	:	English
Software	:	Python (PyTorch and its ecosystem), Jupyter notebooks
Course materials	:	Slides and short handout for the lectures; Jupyter notebooks for the practical sessions

Learning Outcomes

This course aims at introducing artificial neural networks as well as their extension known as deep learning and providing practical skills to tackle a machine learning task using artificial neural networks. Beforehand, real-world applications of deep learning are presented, and the differences between artificial intelligence, machine learning, and deep learning are discussed. The main concepts of an artificial neural network (layers, activation functions, architectures) and how to train one (stochastic gradient descent, batches, momentum) are then defined. The following most common architecture families are presented: multilayer perceptrons, convolutional neural networks, recurrent neural networks, (variational) autoencoders, generative adversarial networks, transformers, and diffusion models. The introduction of each architecture family is motivated by concrete examples of relevant machine learning tasks. During practical sessions, such concrete examples of relevant machine learning tasks will be addressed using artificial neural networks. The materials for the practical sessions are Jupyter notebooks, the programming language that will be used is Python and the Python packages that will be used are PyTorch and its ecosystem.

Prerequisites

Basic linear algebra (vectors, matrices, matrix multiplication), gradient descent, basic Python (object-oriented programming in Python would be a plus but is not required)

Subjects Covered

- Illustrative examples of deep learning
- Differences between artificial intelligence, machine learning and deep learning
- Basic concepts of artificial neural networks: layers, activation functions, architectures
- Training an artificial neural network: stochastic gradient descent and its variants
- Common neural network architecture families: multilayer perceptrons, convolutional neural networks, recurrent neural networks, (variational) autoencoders, generative adversarial networks, transformers, diffusion models
- Applications: computer vision, signal processing, natural language processing

Evaluation

Written Exam + small project

Bibliography

1. GOODFELLOW Ian, BENGIO Yoshua, COURVILLE Aaron. Deep Learning. MIT Press. 2016. https://www.deeplearningbook.org
2. GERON Aurélien, Hands-On Machine Learning with Scikit-Learn and PyTorch. O'Reilly, 2025. https://www.oreilly.com/library/view/hands-on-machine-learning/9798341607972/
3. VIEHMANN Thomas, STEVENS Eli, ANTIGA Luca Pietro Giovanni and HUANG Howard, Deep Learning with PyTorch. O'Reilly, 2026. https://www.oreilly.com/library/view/deep-learning-with/9781633438859/
4. BUDUMA Nithin, BUDUMA Nikhil and PAPA Joe. Fundamentals of Deep Learning. O'Reilly, 2022. https://www.oreilly.com/library/view/fundamentals-of-deep/9781492082170/
5. PyTorch documentation: https://docs.pytorch.org/docs/

UE-MSD01 – Machine Learning – MSD 01.3 - 1st Semester

Dimension Reduction & Self-supervised Representation Learning

Professor	:	Titouan VAYER (INRIA)
ECTS Credits	:	2
Lectures and Tutorials		18 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	36 hrs
Teaching language	:	English
Software & Packages	:	Python
Teaching materials	:	Blackboard

Learning Outcomes

In modern machine learning, data is increasingly high-dimensional (images, text, sensor streams) and increasingly unlabeled, so two questions become central: how to extract a compact, meaningful structure from raw data, and how to learn useful representations without relying on costly human annotation. Dimension reduction addresses the first question, moving from linear projections such as PCA, which capture directions of maximum variance but fail when the underlying structure is nonlinear, to neighborhood-embedding methods such as t-SNE and UMAP, which preserve local similarity structure and are the standard tools for visualizing and exploring high-dimensional datasets. Representation learning addresses the second question: rather than relying on labels, self-supervised methods define pretext tasks directly from the data itself, and this course focuses on contrastive approaches such as SimCLR, together with the alignment and uniformity framework that explains why such methods avoid trivial solutions, and on sparse autoencoders, which learn compact and disentangled features through a reconstruction objective. This course will state the main principles behind these techniques while emphasizing their methodological and algorithmic aspects.

Prerequisites: Basic statistics, linear algebra, and probability; basic notions of neural networks.

Subjects Covered

- Understand the curse of dimensionality and the motivation for both linear and nonlinear dimension reduction.
- Know PCA, its assumptions, and its limitations on nonlinear data.
- Understand neighborhood embedding methods (t-SNE, UMAP) and their algorithmic principles.
- Understand the principles of self-supervised and contrastive representation learning.
- Know the SimCLR framework and the alignment/uniformity theoretical perspective.
- Understand sparse autoencoders as a non-contrastive route to interpretable representations.

Evaluation: Mean {1 project, 1 oral examination}

Bibliography

1. HASTIE T., TIBSHIRANI R., FRIEDMAN J., The Elements of Statistical Learning, Springer, 2nd Edition, 2009.
2. BALESTRIERO R. et al., A Cookbook of Self-Supervised Learning, 2023.

1st Semester

TEACHING UNIT MSD-02: MODELS FOR DEPENDENT DATA

ECTS Credits	: 5
Estimated personal workload (beyond lecture and tutorial time) /	: 60 to 70 hrs
Lectures and Tutorials	: 42 hrs

Learning Objectives of the Teaching Unit

In many modern applications, temporal dependency needs to be considered to build reliable models and to reach a fine prediction accuracy. Typical examples include weather forecasting or predicting the price of a given financial derivative. The first part of the unit presents methods, algorithms to handle time-series data. The second part of the unit is interested in modeling multivariate (potentially high-dimensional) time-series. The aim is to account for the possible interaction between different time series.

Description

The Models for Dependent Data unit includes 2 courses:

1. Machine Learning for Time Series
2. High-Dimensional Time Series

Acquired Skills

Build statistical model taking into account time-dependency in data, estimate the model using dependent data

Prerequisites

Probability theory (covariance matrix, correlation, Gaussian vector), Linear algebra, Basic machine learning algorithms.

UE-MSD02 – Models for Dependent Data – MSD 02.1 - 1st Semester

Machine Learning for Time Series

Professor	:	Simon MALINOWSKI (IRISA)
ECTS Credits	:	2
Lectures and Tutorials	:	18 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	10 hrs
Teaching language	:	English
Software & Packages	:	Python
Teaching materials	:	

Learning Outcomes

When learning from structured data such as time series data, special attention has to be paid to the models used. Indeed, designing machine learning models requires thinking of the invariants to be learned, and either encoding them in the model or designing the model so that it is able to discover such invariants and encode them. In this course, we will cover the use of alignment-based methods in traditional machine learning models. Dedicated neural network architectures will also be tackled. All these models will be illustrated on real datasets. After this course, the student will be able to choose an adequate machine learning model and apply it for a given time series task.

Prerequisites

Basics of neural networks

Subjects Covered

- Shift invariance in time series
- Alignment-based methods for time series
- Shapelets for time series classification
- Recurrent neural networks
- Convolutional models for time series

Evaluation

Report on a real-data analysis. The report will be initiated during class hours.

Bibliography

1. GOODFELLOW, I., BENGIO, Y., COURVILLE, A. (2016). Deep learning. MIT Press, 2016.

UE-MSD02 – Models for Dependent Data – MSD 02.2 - 1st Semester

High-Dimensional Time Series

Professors	:	Lionel TRUQUET (ENSAI)
ECTS Credits	:	3
Lectures and Tutorials	:	24 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	30 hrs
Teaching language	:	English
Software & Packages	:	R
Teaching materials	:	Slides, data sets and R codes on Moodle

Learning Outcomes

Handling multivariate time series, performing a descriptive analysis on time series data, selecting an appropriate model for multivariate time series, utilizing appropriate methods for statistical inference, prediction or classification of multivariate and high-dimensional time series.

Prerequisites

Basics in probability theory, statistical inference, linear algebra. Knowledge of the software R.

Subjects Covered

In this course, we will introduce the primary tools for analyzing time series data. We will begin by presenting the key concepts for dealing with univariate time series, such as trend, seasonality, and stationary processes. Subsequently, we will delve into the main models and inference methods for multivariate linear time series. In the latter part of the course, we will explore the scenario of multiple time series with a substantial number of components. To address high-dimensional parameter spaces, we will introduce the LASSO penalty and its variants as well as low-rank methods. Towards the end of the course, we will provide an introduction to clustering or classification problems in the context of time series analysis. Real-world data examples and the software R will be used to illustrate all the methods.

Evaluation

Project on a real data set with an oral presentation

Bibliography

1. LÜTKEPOHL, H. (2005). New introduction to multiple time series analysis. Springer Science & Business Media.
2. PEÑA, D., & TSAY, R. S. (2021). Statistical learning for big dependent data. John Wiley & Sons.

1st Semester

TEACHING UNIT MSD-03: STATISTICS FOR NEW DATA

ECTS Credits	: 5
Estimated personal workload (beyond lecture and tutorial time)	: 63 hrs
Lectures and Tutorials	: 42 hrs

Learning Objectives of the Teaching Unit

Modern applications produce data under various complex forms. Many existing data are composed of curves, images or are graph-structured, for which standard regression technique or the time-series paradigm is not applicable or relevant. This unit presents, on one hand, the functional data analysis (FDA) approaches and, on the other hand, graphical models to handle such kind of particular data. This unit will hence explore alternative approaches to the classical ones as developed in the Machine Learning and Time-Series units.

Description

There are two courses:

1. Functional Data Analysis
2. Machine Learning for Natural Language Processing

Acquired Skills

Model complex data using advanced methods.

Prerequisites

Analysis and linear algebra, Probability theory

UE-MSD03 – Statistics for New Data – MSD 03.1 - 1st Semester

Functional Data Analysis

Professor	:	Valentin PATILEA (ENSAI)
ECTS Credits	:	3
Lectures and Tutorials	:	24 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	1.5 h personal workload per 1h lecture
Teaching language	:	English
Software & Packages	:	R
Teaching materials	:	Lecture notes and textbook (see below)

Learning Outcomes

This course aims to provide an introduction to functional data analysis (FDA), that is to say to methods for analyzing data generated by curves, surfaces or other types of objects in non-Euclidean spaces. The fundamental statistical tools for modeling and analyzing such data will be explored. This course introduces ideas and methodology in FDA as well as the use of software. Students will learn the idea of different methods and the related theory, and also the numerical and estimation routines to perform functional data analysis. Students will also have an opportunity to learn how to apply FDA to a wide array of application areas. The course will contain several examples where FDA techniques have a clear advantage over classical multivariate techniques. Some recent developments in FDA will also be discussed.

Prerequisites:

Statistical inference and methods, Multivariate statistical analysis, Nonparametric smoothing methods

Subjects Covered

Chapter 1. Introduction.
Chapter 2. Representing functional data and exploratory data analysis. Including: basic expansions, FPCA, derivatives, smoothing, packages.
Chapter 3. Basic elements of Hilbert space theory and random functions.
Chapter 4. Estimation and inference from a random sample. Including, estimation of functional principal component analysis (FPCA). Inference about the mean function.
Chapter 5. Functional Linear regression models. Including: Functional linear regression models with scalar and functional response variable (function-on-scalar, scalar-on-function and function-on-function models).
Chapter 6. (depending on the available time) Functional generalized linear models or Functional time series.

Evaluation:

The final grade will be determined by two criteria: Reading and presenting a research article (40%) and Final exam (60%)

Bibliography

1. CRAINICEANU, C.M., GOLDSMITH J., LEROUX A., CUI E. Functional Data Analysis with R. Boca Raton, FL: Chapman & Hall/CRC Press, 2024.

2. GERTHEISS, J., et al. Functional data analysis: An introduction and recent developments. Biom. J., 66(7): e202300363, 2024.

3. KOKOSZKA, P and REIMHER, M. Introduction to Functional Data Analysis. Chapman & Hall/CRC, Texts in Statistical Science, 2017.
4. RAMSAY, J.O., HOOKER, G. and GRAVES, S. Functional Data Analysis in R and Matlab. Springer, 2009.
5. RAMSAY, J.O. and SILVERMAN, B. W. Functional Data Analysis. Springer, 2005.
6. SHANG, H.L. ftsa: An R package for analysing functional time series. The R journal, 64-72, 2013.

UE-MSD03 – Statistical for New Data – MSD 03.2 - 1st Semester

Machine Learning for Natural Language Processing (NLP)

Professor	:	Guillaume GRAVIER (IRISA)
ECTS Credits	:	2
Lectures and Tutorials	:	18 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	18hrs
Teaching language	:	English
Software & Packages	:	PyTorch, Scikit learn, SpaCy, HuggingFace's libraries
Teaching materials	:	Slideware

Learning Outcomes

- Understand and analyze recent NLP models
- Implement natural language processing pipelines
- Design solutions for text information extraction

Prerequisites

- Foundations of machine learning (probability/statistics, optimization, gradient descent, loss function, etc.)
- Good knowledge of Python
- Familiarity with tensirflow/keras and/or pyTorch

Subjects Covered

The course will introduce the main notions of NLP and detail machine learning based approaches to modern NLP, going through the following: word representation, text classification, word tagging, language modeling, transformers and large language models, text generation.

Evaluation

Evaluation will be based on a short quiz and on personal projects. The latter can be done in groups of two students.

Bibliography

1. DAN JURAFSKY and JAMES H. MARTIN, Speech and Language Processing (3rd ed. draft) https://web.stanford.edu/~jurafsky/slp3/

1st Semester

TEACHING UNIT MSD-04:
ADVANCED TOOLS FOR DATA ANALYSIS & COMPUTING

ECTS Credits	: 3
Estimated personal workload (beyond lecture and tutorial time)	: 20 to 30 hrs
Lectures and Tutorials	: 33 hrs

Learning Objectives of the Teaching Unit

This teaching unit develops two important topics that lie at the heart of any data analysis: data visualization and parallel computing. Those 2 topics are often left aside in most Machine learning or statistical courses (which most often are interested in modeling and predicting). They here have their own place. Data visualization is a set of techniques allowing to summarize visually some piece of information contained in the data but also to allow determining some patterns in the data. Parallel computing consists in sending what needs to be computed to different machines in order to reduce computing time. This is necessary in most large-scale learning problems.

Description

There are two courses:

1. Data visualization
2. Parallel computing with R & Python

Acquired Skills

Algorithm complexity

Prerequisites

Basics on R and Python. A minimal knowledge of the basic tools used in data science, as well as in statistics is required such as: PCA, classification algorithms.

UE-MSD04 – Advanced Tools for Data Analysis & Computing – MSD 04.1 - 1st Semester

Data Visualization

Professor	:	Etienne MADINIER
ECTS Credits	:	1
Lectures and Tutorials	:	15 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	10 hrs
Teaching language	:	English
Software & Packages	:	Python
Teaching materials	:	Slides, exercises

Learning Outcomes

Notions in graphic semiology to be able to choose the relevant vizualisation. Creation of interactive diagrams, cartographic or otherwise, to represent datasets, in Python.

Prerequisites

Basics on Python

Subjects Covered

Data visualization is a fundamental ingredient of data science as it “forces us to notice what we never expected to see” in a given dataset.
Dataviz is also a tool for communication and, as such, is a visual language.
All along the courses, we will focus on methods and strategies to represent datasets, using dynamic and interactive tools.

Evaluation

The evaluation consists on a data vizualisation project. The students will have to build a web site, based on Bokeh library.

Bibliography

1. Official Python documentation :https://docs.python.org/
2. Matplotlib : https://matplotlib.org/
3. Bokeh documentation : https://docs.bokeh.org/en/latest/

UE-MSD04 – Advanced Tools for Data Analysis & Computing – MSD 04.2 - 1st Semester

Parallel Computing & Data Science with R, Julia and Python

Professors	:	Aymeric STAMM (CNRS) - lectures on "R" Emmanuel PILLIAT (ENSAI) - lectures on "Julia & Python"
ECTS Credits	:	2
Lectures and Tutorials	:	18 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	12 to 15 hrs
Teaching language	:	English
Software & Packages	:	R, Julia and Python
Teaching materials	:	Material on Moodle for R, Julia and Python. Julia and Python material on https://epilliat.github.io/teaching/parallel_computing/

Learning Outcomes

- R-Julia-Python: Detecting the slow parts of a script by using profiling and benchmarking tools. Students will be able to detect the parts of a script where the code should be improved and where the memory allocations should be reduced.
- Julia: Understanding why a just-in-time compiled language can specialize code on types, and how this reduces the per-operation overhead, with Python as a familiar point of comparison.
- Julia: Understanding multiple dispatch as a language mechanism, contrasted with the object-oriented style of Python.
- R-Julia-Python: Knowing the various ways of implementing parallel computations.
- Julia & Python: Implementing multithreaded computations on the CPU and recognizing the classic pitfalls, such as race conditions and the limits that some language runtimes place on threading.
- Julia & Python: Distinguishing concurrency from parallelism and using asynchronous tasks to overlap waiting time on a single core.
- Julia: Gaining an awareness of the memory hierarchy and of data locality and, time permitting, an overview of how a computation can be offloaded to a GPU.
- R: Learning and using the futureverse ecosystem of packages, a unifying parallelization framework in R with which you can parallelize locally or on clusters.
- R: Learning and using the mirai and mori packages which offer the same capabilities with different trade-offs and support shared memory access.
- R: Using parallel computing for exploratory data analysis within the tidyverse.
- R: Using parallel computing for tuning ML models within the tidymodels.
- Improving code performance through parallel computation on the CPU and, time permitting, on the GPU.

Prerequisites

Basic knowledge of R and Python. No prior knowledge of Julia is required — it is introduced from scratch, with Python as the point of comparison.

Subjects Covered

In the R section, we will learn how to profile the code to look for slow parts or memory-heavy parts. We will then learn a few tricks to make sure the basic R code is optimized before thinking about parallelization. Next, we will introduce various ways of implementing parallel computations in R with their pros and cons. Finally, we will learn about the futureverse framework which is a unifying framework for parallel computing in R.

We will dive into the core concepts of futures and showcase the recent futurize and progressify packages that make turning sequential code into parallel code frictionless. We will also learn about the mirai and mori packages for parallelization which use different tradeoffs and support shared memory access. The course will then be oriented towards showing how simple it is to use parallelization within the tidyverse ecosystem for exploratory data analysis and within the tidymodels ecosystem for machine learning thanks to the futureverse and the mirai package.

In the Julia & Python section, the guiding idea is that speed does not come from a "magic" language but from understanding what the machine is actually doing. Python serves as the familiar baseline and Julia as the test bench. Through a series of small examples, we look at how code specialized on types differs from interpreted code, how dispatch on types compares with the object-oriented style, and how concurrency differs from parallelism: first overlapping waiting time with asynchronous tasks on a single core, then multithreading and its classic pitfalls, such as race conditions. Timings are compared along the way, so that students can see where the speed-ups actually come from. Depending on the time available, the section may also offer an overview of the usual Python tools for numerical and parallel computing, and of offloading a computation to a GPU. The precise selection of topics and libraries may vary from one year to the next.

Evaluation

A coding project: students present a computation they have profiled and optimized, showing where the speed-ups come from and explaining why — cost per operation, reduced memory allocations, and parallelization. The report is written at home, followed by an oral presentation.

Julia itself is not graded — the language is a means, not an end. What is assessed is the optimization approach: the ability to identify bottlenecks, justify the diagnosis, and explain the reasoning behind each speed-up, regardless of the language used.

A priori same evaluation method for the R part.

Bibliography

1. https://www.futureverse.org/
2. https://journal.r-project.org/archive/2021/RJ-2021-048/RJ-2021-048.pdf
3. https://arxiv.org/pdf/2601.17578
4. https://mirai.r-lib.org
5. https://shikokuchuo.net/mori/
6. https://tidyverse.org/blog/2025/07/purrr-1-1-0-parallel/
7. https://tidyverse.org/blog/2020/11/tune-parallel/
8. https://wiki.python.org/moin/ParallelProcessing
9. https://docs.julialang.org/en/v1/manual/performance-tips/
10. https://docs.julialang.org/en/v1/manual/parallel-computing/
11. https://docs.python.org/3/library/concurrency.html

1st Semester

TEACHING UNIT MSD-05:
IT TOOLS

ECTS Credits	:	5
Estimated personal workload (beyond lecture and tutorial time)	:	35 to 40 hrs
Lectures and Tutorials	:	42 hrs

Learning Objectives of the Teaching Unit

In modern applications, the collected data is often associated with a large dimension (big data's one v is volume) and needs to be treated in a small amount of time (big data's other v is velocity). In such context, IT tools have become of prime importance in data science. This unit presents a panorama of modern computer/cloud tools for processing massive amounts of complex data.

Description

Courses of NoSQL, Hadoop and Spark are proposed.

Acquired Skills

Using the most recent computer/cloud computing tools (Hadoop and Spark) for data processing.

Prerequisites

Basics in programming and databases: Java, Python, R, Linux, SQL.

UE-MSD05 – IT Tools – MSD 05.1 - 1st Semester

IT Tools 1 (Hadoop & Cloud Computing)

Professor	:	Shadi IBRAHIM (INRIA – Rennes)
ECTS Credits	:	2
Lectures and Tutorials	:	18 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	9 to 15 hrs
Teaching language	:	English
Software & Packages	:	Hadoop, Virtual Machine Mangers (e.g., Virtual Box, VMware-Player, VMware Fusion, etc), Docker
Teaching materials	:	All course materials presentations, tutorials and hand-ons, libraries and codes will be available online on the course website in pdf and zip format.

Learning Outcomes

At the end of the lectures, the student will realize the potential of Big Data and will know the main tools to process this tsunami of data at large-scale. In particular, the students will understand the main features of MapReduce programming model and its open-source implementation Hadoop, and will be able to use Hadoop and test it using different configurations.
Data volumes are ever growing, for a large application spectrum going from traditional database applications, scientific simulations to emerging applications including Web 2.0 and online social networks. To cope with this added weight of Big Data, we have recently witnessed a paradigm shift in computing infrastructure through Cloud Computing and in the way data is processed through the MapReduce model. First promoted by Google, MapReduce has become, due to the popularity of its open-source implementation Hadoop, the de facto programming paradigm for Big Data processing in large-scale infrastructures. On the other hand, cloud computing is continuing to act as a prominent infrastructure for Big Data applications.

The goal of this course is to give a brief introduction to Cloud Computing: definitions, types of cloud (IaaS/PaaS/Saas, public/private/hybrid), challenges, applications, main cloud players (Amazon, Microsoft Azure, Google etc.), and cloud enabling technologies (virtualization). Then we will explore data processing models and tools used to handle Big Data in clouds such as MapReduce and Hadoop. An overview on Big Data including definitions, the source of Big Data, and the main challenges introduced by Big Data, will be presented. After that, we will discuss distributed file systems. We will then present the MapReduce programming model as an important programming model for Big Data processing in the Cloud. Hadoop ecosystem and some of major Hadoop features will then be discussed.

Prerequisites

- Familiar with Linux command-line
- Familiar with Java/Python

Subjects Covered

Throughout the course we will cover the following topics:
- Cloud Computing: definitions, types, Challenges, enabling technologies, and examples (2.25 hrs)
- Big Data: definitions, the source of Big Data, challenges (1.5 hrs)
- Google Distributed File System (1.5 hrs)
- The MapReduce programming model (1.5 hrs)
- Hadoop Ecosystem (2.25 hrs)

- Practical sessions on Hadoop (7 hrs)
  - ✓ How to use Virtual Machines/Containers and Public Cloud Platforms
  - ✓ Starting with Hadoop
  - ✓ Configuring HDFS
  - ✓ Configuring and Optimising Hadoop
  - ✓ Writing MapReduce applications

Independent work (tentative): Students will be divided into groups where each group will do a 15 - 20 min presentation on one of the main subjects or a life demonstration on one of the practical sessions (2 hrs)

Evaluation

Written exam

Bibliography

1. JIN Hai, IBRAHIM Shadi, BELL Tim, GAO Wei, HUANG Dachuan, WU Song. Cloud Types and Services. Book Chapter in the Handbook of Cloud Computing, Springer Press, 26 Sep 2010.
2. JIN Hai, IBRAHIM Shadi, BELL Tim, LI QI, HAIJUN Cao, WU Song, XUANHUA Shi. Tools and technologies for building the Clouds. Book Chapter in Cloud Computing: Principles Systems and Applications, Springer Press, 2 Aug 2010.
3. ARMBRUST Michael, FOX Armando, GRIFFITH Rean,. JOSEPH Anthony D, KATZ Randy, KONWINSKI Andy, LEE Gunho, PATTERSON David, RABKIN Ariel, STOICA Ion, and ZAHARIA Matei. 2010. A view of cloud computing. Commun. ACM 53, 4 - April 2010.
4. GHEMAWAT Sanjay, GOBIOFF Howard, and LEUNG Shun-Tak. The Google file system. In SOSP '03.
5. DEAN Jeffrey, GHEMAWAT Sanjay, OSDI, MapReduce: Simplified Data Processing on Large Clusters. 2004.
6. JIN Hai, IBRAHIM Shadi, LI QI, HAIJUN Cao, WU Song, XUANHUA Shi. The MapReduce Programming Model and Implementations. Book Chapter in Cloud Computing: Principles and Paradigms.
7. VAVILAPALLI Vinod Kumar, MURTHY Arun C., DOUGLAS Chris, AGARWAL Sharad, KONAR Mahadev, EVANS Robert, GRAVES Thomas, LOWE Jason, SHAH Hitesh, SETH Siddharth, SAHA Bikas, CURINO Carlo, O'MALLEY Owen, RADIA Sanjay, REED Benjamin, and BALDESCHWIELER Eric. Apache Hadoop YARN: yet another resource negotiator. In SOCC '13.

UE-MSD05 – IT Tools – MSD 05.2 - 1st Semester

IT Tools 2 (NoSQL, Big Data Processing with Spark)

Professors	:	Nikolaos PARLAVANTZAS (IRISA Rennes) - NoSQL Hervé MIGNOT (Equancy) – Big Data Processing with Spark
ECTS Credits	:	3
Lectures and Tutorials (total)	:	24 hrs
Teaching language	:	English

NoSQL

Professor	:	Nikolaos PARLAVANTZAS (IRISA Rennes) - NoSQL
Lectures and Tutorials	:	9 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	10 hrs
Software & Packages	:	Redis, Elasticsearch, Cassandra, Neo4j
Teaching materials	:	Slides and lab subjects on Moodle

Learning Outcomes

Understand the fundamentals of NoSQL databases and the features and specific challenges NoSQL databases are addressing compared to classic SQL databases. Evaluate and select appropriate NoSQL technologies for particular situations. Gain hands-on experience in deploying and using NoSQL databases, such as MongoDB or Neo4j.

Prerequisites

Basic knowledge of SQL, databases, and computer systems

Subjects Covered

- NoSQL origins (history & players)
- NoSQL / SQL comparison
- Key concepts of NoSQL databases:
  - ✓ Data models
  - ✓ Distribution models
  - ✓ Query languages
  - ✓ Consistency
- NoSQL database types
- NoSQL database technologies & comparisons (MongoDB, Cassandra, Neo4j, Redis, ElasticSearch…)
- Neo4j introduction + lab
- Cassandra introduction + lab

Evaluation: Project

Bibliography

Many online resources are available

Big Data Processing with Spark

Professor	:	Hervé MIGNOT (Equancy)
Lectures and Tutorials	:	15 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	14 hrs
Software & Packages	:	Spark
Teaching materials	:	Notes & scripts

Learning Outcomes

Understand the stakes of distributed computing through the Apache Spark architecture. Discover how to use Apache Spark, platforms & tools available. Practice PySpark coding to learn Apache Spark features, from data management to machine learning.

Prerequisites

Computer systems and architecture basic knowledge, Python & SQL language practice

Subjects Covered

- Distributed computing introduction
- Apache Spark origins & history, links to Apache Hadoop
- Apache architecture and main concepts:
  - Apache Spark “modules”
  - Architecture: driver & executors
  - Transformations vs. actions
  - Lazy evaluation
  - Data structures: RDD, dataframes & datasets
- Using Apache Spark:
  - ✓ Create sessions and connect to clusters
  - ✓ Use data management functions
  - ✓ Leverage SQL with Spark SQL
  - ✓ Train & test machine learning models
- Use Spark Web UI

Evaluation

Questionnaire and Project

Bibliography

1. Apache Spark online documentation: https://spark.apache.org/docs/latest/
2. KARAU H., WARREN R. High Performance Spark (2017) – 2nd edition (2026)

1st Semester

TEACHING UNIT MSD-06: CASE STUDIES & PROJECT

ECTS Credits	: 5
Estimated personal workload (beyond lecture and tutorial time)	:
Lectures and Tutorials	: 60 hrs

Learning Objectives of the Teaching Unit

This teaching unit has been organized to offer the students the opportunity to work on new subjects using the knowledge they acquired during the first semester. This is an important step toward completing the master's program as the students should demonstrate their ability to draw some links between the previous teaching units in order to discover new topics. The first part consists in working on a project as a team (two or three by team, supervised by a field expert) and the second part is divided into multiple seminar sessions (each is dedicated to a recent data science topic).

Description

There are two phases in this teaching unit:

1. The project
2. The seminars

Acquired Skills

Knowledge on some specific hot-topics in data science and the ability to work as a team on a (research-like) project

Prerequisites

All previous courses given in the Master.

UE-MSD06-Case Studies and Project – MSD 06.1 - 1st Semester

Smart Data Project or Research Project

Supervisors : Several industrial or lab partners
ECTS Credits : 2.5

Learning Objectives

The main part of courses focuses on studying several facets of statistics, mathematics and computer sciences, according to the Big/Smart Data paradigm. One of the main objectives of this project is to apply this new knowledge learned during the 1st semester into a unique application. This project puts into practice theoretical methods studied in different courses and starts with project management.

The learning objective is not limited to putting the theory learned in other courses into practice but aims to raise awareness of other aspects linked to project management among students, such as communication (between students and with the client that proposed the project).

This project should provide additional support, be carried out by an expert in the field, according to the needs of students. The expert is expected to provide

- Supervising at start for requirement
- Distant supervising on technical queries
- Technical supervising during implementation phase
- Help for defense preparation

Main Subjects covered

The topic of the Smart Data project could be related to any type of application requiring advanced data science tools.

Evaluation

The evaluation is two-fold:

1 - a report written by all students of each project team, eventually supervised by the external organism.
2 - a project defense in front of a jury

UE-MSD06-Case Studies and Project – MSD 06.2 - 1st Semester

Topics, Case Studies, Conferences /or Research Project

Professors		Romaric GAUDEL(IRISA) Shadi IBRAHIM (INRIA - Rennes) Rémi LELUC (Qube Research & Technologies) Thomas ZAMOJSKI (Quadratic) Franck ORAGA (Exor Data, Consultant Hso-Microsoft)
ECTS Credits	:	2.5
Lectures and Tutorials (total)	:	36 hrs (Ensai)
Teaching language	:	English

Bandit Theory

Professor	:	Romaric GAUDEL (IRISA)
Lectures and Tutorials	:	6 hrs
Estimated personal workload (beyond lecture and tutorial time)	:	3 hrs
Software & Packages		Jupyther notebook
Teaching materials	:	Slides & notebook

Learning Outcomes

You will:
- learn how to identify when exploration is necessary in a learning system
- learn standard strategies for handling this requirement
- implement and test (with notebooks) these strategies.

The need for exploration is ubiquitous in applications, arising as soon as we learn a model from data resulting from the choices made by that model. This challenge is one of the fundamental obstacles in reinforcement learning and recommender systems.

Prerequisites

Basic knowledge of Python and of object-oriented programming
Basic knowledge of Machine Learning would be a plus

Subjects Covered

Bandit setting and use-cases
Analysis of Explore then Commit strategy
Presentation, implementation and test of standard solutions: epsilon-greedy, UCB, Thompson Sampling

Evaluation:

A quizz at the end of the session
Evaluation of the notebooks completed during practical sessions

Bibliography

1. MYLES WHITE John. Bandit algorithms for Website optimization. O'Reilly Media.
2. LATTIMORE Tor and SZEPESVARI Csaba. Bandit Algorithms. Cambridge University Press
3. BUBECK Sébastien. CESA-BIANCHI Nicolo. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. Foundations and Trends in Machine Learning, Vol. 5, No. 1 (2012) 1–122.

Some Recent Advances for Big Data Processing in the Cloud

Professor	:	Shadi IBRAHIM (INRIA – Rennes)
Lectures and Tutorials	:	6 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	3 to 5 hrs
Software & Packages	:	Hadoop
Teaching materials	:	All course materials presentations, tutorials and hand-ons, libraries and codes will be available online on the course website in pdf and zip format.
	:	

Learning Outcomes

At the end of the lectures, the student will be able to identify the main performance bottlenecks when running Big data applications in Clouds and will know how the performance of Hadoop can be improved, accordingly. During this conference, we will discuss several approaches and methods used to optimize the performance of Hadoop in the Cloud. We will also discuss the limitations of Hadoop and introduce state-of-the-art resource management systems and job schedulers for Big data applications including Mesos, Delay scheduler, ShuffleWatcher, and Tetrium. In addition, we will discuss how redundancy techniques, such as replication and erasure coding, affect the performance of MapReduce applications.

Prerequisites

Attend the course: Big Data processing in Clouds: Hadoop

Subjects Covered

Approaches to optimize Hadoop in clouds (2.5 hrs)
Resource management and job scheduling for Big data applications: Mesos, Delay scheduler, ShuffleWatcher, Tetrium, etc (2.5 hrs)

Independent work (tentative): Students will be assigned to groups where each group will do a 15 -20 min presentation (1 hr)

Evaluation

During the session and/or a technical report to be submitted after the session

Bibliography

1. Apache Hadoop YARN: yet another resource negotiator. VAVILAPALLI Vinod Kumar, MURTHY Arun C., DOUGLAS Chris, AGARWAL Sharad, KONAR Mahadev, EVANS Robert, GRAVES Thomas, LOWE Jason, SHAH Hitesh, SETH Siddharth, SAHA Bikas, CURINO Carlo, O'MALLEY Owen, RADIA Sanjay, REED Benjamin, and BALDESCHWIELER Eric. In SOCC '13.

2. IBRAHIM Shadi, PHAN Tien-Dat, CARPEN-AMARIE Alexandra, CHIHOUB Houssem-Eddine, MOISE Diana, ANTONIU Gabriel. Governing energy consumption in hadoop through cpu frequency scaling: An analysis. In FGCS 2016.

3. PHAN Tien-Dat, IBRAHIM Shadi, ANTONIU Gabriel, BOUGE Luc. On Understanding the energy impact of speculative execution in Hadoop. In GreenCom2015.

4. YILDIZ Orcun, IBRAHIM Shadi, ANTONIU Gabriel. Enabling fast failure recovery in shared Hadoop clusters: Towards failure-aware scheduling. In FGCS 2016.

5. HINDMAN Benjamin, KONWINSKI Andy, ZAHARIA Matei, GHODSI Ali, JOSEPH Anthony D., KATZ Randy, SHENKER Scott, and STOICA Ion. Mesos: a platform for fine-grained resource sharing in the data center. In NSDI'11.

6. ZAHARIA Matei, BORTHAKUR Dhruba, SEN SARMA Joydeep, ELMELEEGY Khaled, SHENKER Scott, STOICA Ion. Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling. In EuroSys'10.

7. ZAHARIA Matei, KONWINSKI Andy, JOSEPH Anthony D., KATZ Randy, STOICA Ion. Improving MapReduce performance in heterogeneous environments. In OSDI'08.

8. AHMAD Faraz, CHAKRADHAR Srimat T., RAGHUNATHAN Anand, VIJAYKUMAR T. N. Shufflewatcher: Shuffle-aware scheduling in multi-tenant mapreduce clusters. In USENIX ATC 2014

9. HUNG Chien-Chun, ANANTHANARAYANAN Ganesh, GOLUBCHIK Leana, YU Minlan, and ZHANG Mingyang. 2018. Wide-area analytics with multiple resources. In EuroSys '18.

10. IBRAHIM Shadi and DARROUS Jad. 2025. Erasure Coding Aware Block Placement for Data-Intensive Applications. In Proceedings of the 5th Workshop on Challenges and Opportunities of Efficient and Performant Storage Systems (CHEOPS '25). Association for Computing Machinery, New York, NY, USA, 15–22.

11. DARROUS Jad, IBRAHIM Shadi and PEREZ Christian, "Is it Time to Revisit Erasure Coding in Data-Intensive Clusters?," 2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Rennes, France, 2019, pp. 165-178, doi: 10.1109/MASCOTS.2019.00026.

Stochastic Optimization Methods for Machine Learning

Professor	:	Rémi LELUC (Qube Research & Technologies)
Lectures and Tutorials	:	6 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	2 hrs
Software & Packages	:	Python
Teaching materials	:	Slides, lecture notes, lab subject and code for practical session

Learning Outcomes

At the end of the lecture, the students will have acquired:

- a robust understanding of theory and applications of stochastic optimization methods.
- a large overview of different stochastic optimization techniques such as Adam, Adagrad, (L)BFGS and general conditioning methods.
- practical techniques to apply stochastic optimization methods to real-world machine learning problems.

Prerequisites

Convex analysis, Linear algebra, Python (basics/numpy/pytorch)

Subjects Covered

This seminar delves into stochastic optimization methods tailored for machine learning, immersing students in both theory and application. The spotlight is on the widely-used Stochastic Gradient Descent (SGD) algorithm and its variants. Exploring the theory behind SGD, we uncover its limitations and expand into enhancements such as diagonal scaling, second-order techniques, and broader conditioning methods. Complementing this, the lecture transitions into a practical session, unraveling the direct application of stochastic optimization in reinforcement learning through policy gradient methods.

Evaluation

Written Exam (QCM) 80% of grade
Lab session (Jupyter notebook) 20% of grade

Bibliography

1. HASTIE Trevor, TIBSHIRANI Robert, FRIEDMAN Jerome. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition Springer Series in Statistics. https://hastie.su.domains/Papers/ESLII.pdf

2. BOTTOU Léon, E. CURTIS Frank, NOCEDAL Jorge. Optimization Methods for Large-Scale Machine Learning, 2018. https://arxiv.org/pdf/1606.04838.pdf

3. SUTTON R.S and BARTO A.G. Reinforcement learning: An introduction, MIT press, 2018 https://www.andrew.cmu.edu/course/10-703/textbook/BartoSutton.pdf

Case Studies in Smart Data
MLOps: Machine Learning in a production environment

Professor	:	Thomas ZAMOJSKI (Quadratic)
Lectures and Tutorials	:	6 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	1.5 – 2 hrs
Software & Packages	:	Python, Docker
Teaching materials	:	

Learning Outcomes

At the end of the lecture, the student will know:
- What are the challenges in deploying and maintaining a machine learning model in operation.
- What are some best practices addressing these concerns.
- How to create a Docker image and run a container.
- How to serve a model as a service in python.
- Statistical methods for online and offline model monitoring.

Prerequisites

Basic knowledge of Python Programming Language

Subjects Covered

Machine Learning models are notoriously hard to put and maintain in production. But why is it so and what can we do about it?
In this course, we will explore the very latest trends in MLOps. We will learn about technologies such as Docker containers and FastAPI. We will also learn statistical methods to intelligently automate model monitoring and we will see how to put them in action via implementations in python packages such as scikit-multiflow and ruptures.

Evaluation

- 60% In-class exercises,
- 30% Code quality and clarity,
- 10% Participation.

Bibliography

1. TRUONG C., OUDRE L., VAYATIS N., Selective review of offline change point detection methods, Signal Processing, September 2019.

2. JAMES N.A., KEJARIWAL A., MATTESON D.S., Leveraging cloud data to mitigate user experience from ‘breaking bad’, 2016 IEEE International Conference on Big Data (Big Data).

3. WEB REFERENCES - 12 factors app: https://12factor.net

Power BI

Professor	:	Franck ORAGA (Exor Data, Consultant Hso-Microsoft)
Lectures and Tutorials	:	12 hrs (ENSAI)
Estimated personal workload (beyond lecture and tutorial time)	:	3 –4 hrs
Software & Packages	:	Power BI Service (online version only) Databricks Free Edition (students must create an account)
Teaching materials	:	Course materials (PDF, slides) Exercise datasets (Retail Purchase Order, HR, etc.) Practical guides (Power Query Online, basic DAX, visualizations) Practical guide: ETL with Python (Pandas) on Databricks Free Edition Power BI Service platform (report sharing)

Learning Outcomes

At the end of the lecture, students will be able to:

- Understand the role of Power BI in the business intelligence ecosystem.
- Get an overview of Power Query Online within Power BI Service.
- Perform ETL data transformations in Python (Pandas) on Databricks Free Edition, including creating and configuring a Databricks account.
- Design a simple data model (relationships, fact and dimension tables).
- Create basic measures with DAX (calculations, aggregations, filters).
- Build and customize an interactive report (dashboards, KPIs, filters).
- Publish and share a report on Power BI Service.
- Work in a team on a practical case study and present a dashboard.

Prerequisites

- Basic knowledge of database modeling (relationships, tables, keys, etc.).
- Fundamental understanding of descriptive statistics and algorithmic logic.
- Basic notions of Python (Pandas) are recommended.
- No prior experience with Power BI is required.

Subjects Covered

- Introduction to Business Intelligence and Power BI
- Power BI Components: Power BI Service (online version only)
- Overview of Power Query Online
- Data Import and ETL Transformation with Python (Pandas) on Databricks Free Edition (account creation, notebooks, data preparation)
- Data Modeling: relationships, star schema (in Power BI Service)
- Introduction to DAX: measures and calculated columns
- Interactive Visualizations: charts, maps, KPIs, filters
- Layout and Data Visualization Best Practices
- Publishing and Sharing on Power BI Service
- Practical Group Cases using real-world data (ETL via Databricks/Pandas → visualization in Power BI Service)

Evaluation: Group project: practical case study analysis + oral presentation

Bibliography

1. Microsoft Fabric, The Complete Guide
2. Business Intelligence with Excel, Power BI, and Office 365
3. Official Microsoft Power BI Documentation: https://learn.microsoft.com/power-bi
4. Additional Resources: SQLBI blog (https://www.sqlbi.com), Microsoft Learn
5. Databricks Documentation: https://docs.databricks.com
6. Pandas Documentation: https://pandas.pydata.org/docs

Second Semester

2nd Semester

TEACHING UNIT MSD-07: INTERNSHIP

ECTS Credits : 30

Working time : Full time internship, for a period of 4 to 6 months

Learning Objectives of the Teaching Unit

The internship is the main bridge between, on one hand, the scientific courses, tutorials and labs and, on the other hand, the world of work. It has two major objectives. First, consolidate students' ability to choose appropriate models, algorithms and computer resources to address real data applications and case studies, to realize proof of concepts and/or develop user solutions, and, finally, explain and provide appropriate arguments for the choices made. Second, place the students in total immersion in a professional environment, in autonomy, as part of a team, in interaction with specialists from the same or complementary fields.

Description

The MSc students are expected to work on topics defined in the internship agreement, under the supervision of a senior professional from the internship unit (private or public company, labs, research institutes…). Each MSc student will have an Ensai adviser who can be contacted for advice.

Acquired Skills

Become a highly skilled specialist in data science able to address complex tasks using up to date modeling tools and computer resources.

Prerequisites

Complete the previous teaching units from the Master's program.

UE-MSD07 - Internship – MSD07.1

End-of-Studies Internship

4-6 months from March to August

Objectives

This final phase of the Master for Smart Data Science program involves a four to six-month paid internship, which can take place either in France or abroad, in either the professional world or academic/research laboratories.

Students should be proactive and begin the search for an internship as early as possible to increase the chances of finding an interesting and relevant internship. Finding an internship is the exclusive responsibility of the student. ENSAI provides assistance in the search process.

This experience should allow for the student to apply the data-science and computer science theory and methods that they have learned during the 1st semester of coursework. Internship topics that are exclusively or almost exclusively oriented towards computer science tools will not be accepted.

The internship should allow students to meet at least two objectives:

- A technical objective: a task is given and, applying theoretical knowledge and skills, the student attempts to complete the task using to the best of his/her ability the resources at his/her disposal.
- A professional objective: the student is immersed in a professional context and must use the internship period to become more knowledgeable and at ease in such an environment, developing professional and personal skills to become a part of the team.

Evaluation

During the internship, students will write a master's thesis that will be examined by the jury and defended by the student in September.