

Computational Learning

Code: 104361
ECTS Credits: 6

2025/2026

Degree	Type	Year
Data Engineering	OB	3

Contact

Name: Ramon Baldrich Caselles
Email: ramon.baldrich@uab.cat

Teaching groups languages

You can view this information at the [end](#) of this document.

Prerequisites

It is essential to have acquired a good mathematical basis as well as to have a good level of programming, mainly in Python.

Objectives and Contextualisation

The course aims to introduce the concepts of artificial intelligence that is based on obtaining knowledge, concepts and trends from the data. It is about training the student to be a "data engineer," and it is one of the professions with the most future and most in demand today by large companies and technology start-ups. In fact, the growth in demand for this data engineering professional is expected to be exponential at the European level, mainly due to the growth in mass data generation. Thus, the main objective of the subject is that the student knows how to find a good solution (sometimes the best is impossible) to problems in different contexts of the treatises, from identifying the needs of representation of the knowledge and, according to this, apply the most appropriate technique (s) to automatically generate good mathematical models that explain the data with an acceptable error.

The contents chosen for this subject are the techniques and concepts that are used extensively in the industry, understanding it in its broadest concept. The algorithmic basis will be fundamental during the development of the subject that wants to have an eminently engineering approach, focusing on the use of the proposals without leaving aside the understanding of the mathematical foundations that support them. The algorithms and techniques shown are the fundamental basis for 'traditional' computational learning without which one cannot understand the techniques that will be developed in future courses. Not because they are basic, they are obsolete, on the contrary, they cover a wide range of applications and problems where they are fundamental. The student must be aware that this knowledge that is the spearhead of the state of the art has an inherent difficulty, involving considerable study and dedication, quantified in hours in the section of formative activities of this guide. . This is because in this subject not only some of the most important contents in the field of machine learning to become a data engineer are taught, but also a curriculum line is worked that allows to expand the range of jobs to which you can access after the degree, as well as lay the methodological bases necessary to do a Master in data engineering or artificial intelligence.

The objectives of the subject can be summarized in:

Knowledges:

- Describe basic computer learning techniques.
- List the essential steps of the different learning algorithms
- Identify the advantages and disadvantages of the learning algorithms that are explained.
- Solve computational problems applying different learning techniques to find the optimal solution.
- Understand the result and limitations of learning techniques in different case studies.
- Know how to choose the most appropriate learning algorithm to solve contextualized problems.

Abilities:

- Recognize situations in which the application of computational learning algorithms may be appropriate to solve a problem
- Analyze the problem to be solved and design the optimal solution applying the techniques learned
- Write technical documents related to the analysis and solution of a problem
- Program the basic algorithms to solve the proposed problems
- Evaluate the results of the implemented solution and assess possible improvements
- Defend and argue the decisions taken in solving the proposed problems

Competences

- Analyse data efficiently for the development of smart systems with the capacity for autonomous learning and/or data mining.
- Students must be capable of collecting and interpreting relevant data (usually within their area of study) in order to make statements that reflect social, scientific or ethical relevant issues.
- Work cooperatively in complex and uncertain environments and with limited resources in a multidisciplinary context, assuming and respecting the role of the different members of the group.

Learning Outcomes

1. Choose the search algorithm and programming paradigm for a problem of optimisation of parameters or states
2. Decide on the most suitable data-learning method depending on the characteristics of the data to be analysed
3. Students must be capable of collecting and interpreting relevant data (usually within their area of study) in order to make statements that reflect social, scientific or ethical relevant issues.
4. Work cooperatively in complex and uncertain environments and with limited resources in a multidisciplinary context, assuming and respecting the role of the different members of the group.

Content

UNIT 1: INTRODUCTION

1.1 Basic concepts and bioinspired paradigms

1.2 History of computer learning

UNIT 2: REGRESSION AND CLASSIFICATION

2.1 Regression of numerical data: gradient descent

2.2 Regularization and logistic regression

2.3 Classification of numerical data: support vector machines

2.4 Decision trees, Random forest

2.5 Bayesian Classification

UNIT 3: CLUSTERING AND SEARCH

3.1 Memorization: lazy learning

3.2 Recommender systems: Content-based vs. Collaborative filtering

3.3 Clustering: k-means and Expectation-Maximization

Activities and Methodology

Title	Hours	ECTS	Learning Outcomes
<hr/>			
Type: Directed			
<hr/>			
Theoretical content	22	0.88	2, 1, 3
<hr/>			
Type: Supervised			
<hr/>			
lab practicums	16	0.64	2, 1, 3, 4
<hr/>			
seminars	10	0.4	2, 1
<hr/>			
Type: Autonomous			
<hr/>			
Setup and development of practical projects	52	2.08	2, 1, 3, 4
<hr/>			
study	28	1.12	2, 1, 3
<hr/>			

All course information and related documents that students may need will be available on the Virtual Campus page (<http://cv.uab.cat/>).

The different activities carried out in the course are organized as follows:

Theory classes

The main concepts and algorithms of each theoretical topic will be presented. These topics serve as the starting point for the course work.

Laboratory sessions

These will be classes where interaction with students is prioritized. They will be individual in nature, although the work can be developed in groups. In these sessions, practical cases will be proposed that require designing a solution using the methods covered in the theory classes. It is not possible to follow the problem-solving sessions without having followed the theoretical content. The outcome of these sessions will

be the resolution of problems, which will be assessed weekly through online tests. The specific mechanism for conducting the assessments will be indicated on the course website. All laboratory sessions will be practical and will include programming a solution to the proposed problem.

Group projects

Work groups will consist of 3-4 students. These groups must remain the same throughout the course and be self-managed: role distribution, work planning, task assignment, resource management, conflict resolution, etc. Although the instructor will guide the learning process, their involvement in group management will be minimal.

Once the material has been presented to understand the challenges of various problems, the problems to be solved will be introduced, and students will define their own project. Throughout the semester, students will work in cooperative groups, analyze the chosen problem, design and implement solutions based on different machine learning algorithms covered in class, analyze the results obtained with each method, and publicly defend their project.

To develop the project, groups will work autonomously, and follow-up sessions will be used to evaluate the work done between sessions and to resolve doubts with the instructor, who will monitor the project's progress, point out errors, suggest improvements, etc. It is essential that groups attend tutorials to receive effective feedback for improving the project. In these sessions, groups must explain the work done, and the instructor will ask questions to all members to assess their contribution. Attendance at these sessions is mandatory.

In the final session of each project, groups will give a presentation explaining the developed project, the adopted solution, and the results obtained. Each group member must participate in the presentation.

Both the theory assessment and the group work will be recoverable.

Annotation: Within the schedule set by the centre or degree programme, 15 minutes of one class will be reserved for students to evaluate their lecturers and their courses or modules through questionnaires.

Assessment

Continous Assessment Activities

Title	Weighting	Hours	ECTS	Learning Outcomes
Individual test	30	7	0.28	2, 1, 3
Problem porfolio	10	5	0.2	2, 1, 3
Project defence	20%	5	0.2	1, 3, 4
Team project	40	5	0.2	2, 1, 3, 4

Activities and Assessment Tools:

This course does not offer a single-assessment option.

To evaluate the acquisition of knowledge and competencies, the assessment combines content assimilation, problem-solving skills, and, significantly, the ability to generate computational solutions to complex problems, both individually and in groups.

The assessment is divided into three parts:

- Content Evaluation

The final content grade will be calculated from several partial exams:

$$\text{Content Grade} = 1/N * \text{Test}_i$$

The number of tests may vary and will be defined at the beginning of the course. To receive a content grade, each test must be graded above 4.

These tests will be conducted during the course and will focus on conceptual understanding of the theoretical sessions.

They aim to individually assess the student's understanding and conceptualization of the techniques taught.

Recovery tests: If the content grade is insufficient, students may take the official exam to retake the failed parts.

No validation of previously passed theoretical parts is allowed.

– Evaluation of laboratory work

The goal of the problem-solving sessions is to engage students continuously with the course content through small exercises that apply theory. Weekly tests will serve as evidence of this work. After each test, students will have access to solutions for self-assessment. Combined with tutoring hours, this helps identify weaknesses.

– Group project evaluation

In the final weeks of the semester, a more extensive project will be carried out. It will be evaluated both as a group and individually. Evaluation criteria include code, report, presentation, and project follow-up during assigned sessions.

Final course grade:

$$\text{Final Grade} = (0.3 * \text{Content}) + (0.1 * \text{Problem Portfolio}) + (0.6 * \text{Project})$$

The project will be graded on both its defense and the quality of its development.

Conditions to pass the course:

- Content grade must be ≥ 4 .
- Project and its defense must be ≥ 6 .

If the calculated final grade is above 5 but the minimums are not met, the final grade will be 4.5.

Honors will be awarded according to current regulations, for grades above 9. In case of ties, additional activities may be proposed.

A student will be marked as "Not Assessable" if no part of the course (theoretical or practical) has been evaluated.

Each grade release will include instructions for recovery if applicable.

Important notices:

- Continuous assessment dates and deadlines, as well as all teaching materials, will be published on the Virtual Campus (<http://cv.uab.cat/>) and may be subject to changes. All updates will be communicated via the Virtual Campus.
- For each assessment activity, a review date, time, and location will be provided. If the student does not attend, no further review will be allowed.
- Irregularities (copying, allowing copying, plagiarism, use of unauthorized devices, etc.) will result in a grade of 0 and cannot be recovered. If these activities are required to pass, the course will be failed. The transcript grade will be the lower of 3.0 or the weighted average.

Bibliography

Web links

- Subject web page: <http://cv.uab.cat>
- Artificial Intelligence: A Modern Approach. <http://aima.cs.berkeley.edu/>

Basic Bibliography

- S. Russell, P. Norvig. Artificial Intelligence: A Modern Approach. Ed. Prentice Hall, Second Edition, 2003. (Existeix traducció al castellà: Inteligencia artificial: Un Enfoque Moderno)
- T. Mitchell. Machine Learning. McGraw Hill. 1997.

Additional Bibliography

- C. Bishop. Pattern Recognition and Machine Learning. Springer-Verlag New York, Inc. 2006

Software

No special software other than the usual ones will be used in these studies.

Groups and Languages

Please note that this information is provisional until 30 November 2025. You can check it through this [link](#). To consult the language you will need to enter the CODE of the subject.

Name	Group	Language	Semester	Turn
(PAUL) Classroom practices	81	Catalan	first semester	afternoon
(PAUL) Classroom practices	82	Catalan	first semester	afternoon