Programming for Data Science
Data Science and Advanced Analytics

Teaching Staff

Flávio L. Pinheiro (FLP) (fpinheiro@novaims.unl.pt) has been a professor of Data Science at NOVA IMS since 2018, where he arrived after a two-year Postdoctoral appointment at the MIT Media Lab (USA). His work explores the role of social networks in the diffusion of opinions, ideas, innovations, and behaviors and how the network structure of socioeconomic systems constrains the effectiveness of strategic decision-making. He holds a Ph.D. in Physics from Universidade do Minho (2016) and an MSc (2011) and BSc (2009) from Faculdade de Ciências da Universidade de Lisboa.

Liah Rosenfeld (lrosenfeld@novaims.unl.pt) is an Adjunct Lecturer in Data Science and a Ph.D. student at Nova IMS. Liah holds a BSc in Psychology (ISCTE), and a master's in Data Science and Advanced Analytics (Nova IMS). Her research aims to improve the learning algorithms that train supervised models using evolutionary computation techniques such as Genetic Programming and Geometric Semantic Genetic Programming.

Niclas Sturm (nfsturm@novaims.unl.pt) is a Ph.D. student in Information Management at NOVA IMS. He holds a BSc in Economics and Ancient History (50%-50%) (University of Heidelberg/Germany) and an MSc in Business Analytics (NOVA SBE). His research focuses on understanding network phenomena in public procurement markets.

Maria Almeida (malmeida@novaims.unl.pt) is an Invited Teaching Assistant in Data Science at NOVA IMS. Maria holds a BSc in Information Management (NOVA IMS) and an MSc in Engineering and Data Science (IST). She currently works as a Data Scientist at Neuraspace, a portuguese startup for space traffic management. Her main interests relate to applying machine learning models for solving day-to-day problems.

Contact

All communications with the instructors can be done through Moodle or via e-mail. Questions regarding the homeworks must be posted on Moodle in the corresponding forum.

Office Hours

This Curricular Unit doesn't have scheduled office hours. Instead, office hours will be scheduled on a per-demand basis, depending on the availability of the teaching staff and the student. If you want to schedule a meeting with a teacher, send us an e-mail.

Description/General Objectives

The *Programming for Data Science* curricular unit is aimed at students who have yet to gain prior programming experience. In this unit, students will learn the fundamentals of programming in Python necessary for a successful career in data science. Starting from the basics of programming, we will rapidly evolve towards advanced computing techniques and concepts of interest for developing a data science project. During the *Programming for Data Science* curricular unit, students will acquire experience working with the backbone stack of libraries (Pandas, Numpy, Scipy, Seaborn, Statsmodels, NetworkX) that make Python the language of choice among data scientists.

At the end of the curricular unit, students are expected to have the capacity to use programming to develop a data science project independently and to feel comfortable with the programming activities in other curricular units. The curricular unit has a strong, active learning component, and, as such, students are expected to participate during classes and read the recommended weekly materials.

Enrollment Requirements

The curricular unit does not have technical enrollment requirements.

Classes will be taught in English, and as such, students are expected to have good comprehension and communication in English.

Intended Learning Outcomes

LO1 - Explain why Python is the preferred programming language for Data Scientists;
LO2 - Understand the basics of the Python programming language;
LO3 - Explain what a Python program does;
LO4 – Develop the ability to independently identify bugs in your code and find the appropriate solutions;
LO5 - Identify and **Use** adequate libraries for your data science needs;
LO6 - Perform the extraction, manipulation, analysis, modeling, and reporting of data using Python;
LO7 - Develop simple Python programs to support your data science projects;
Feel Comfortable using Python!

Syllabus

The curricular unit is organized into three Learning Units (LU):
LU0. Introduction to programming fundamentals using Python
LU1. Exploration of the most relevant libraries in the Python data science stack.
LU2. Use the entire stack and its parts to develop a data science project.

Curricular Unit Planning

The following is a draft of the curricular unit plan. Changes to the proposed plan can be made.

Week	Lectures	Lab
1	UC Program Overview Why Python? What is a computer program? IPython and Shell Readings: Ref1 Part I.1 and I.2 | Ref2 Chap 1, and Appendix B	Software setup Support, Variable types and data structures in Python, Prints
2	Language Semantics Variables and Data Structures Operators, Typecast, Slices, and Comprehensions Flow Control (Loops and If/Else Statements) Readings: Ref2 Chap 2 and Chap 3.1	Flow Control: loops and if statements List comprehensions
3	Functions Import Modules and Python STL NameSpace Readings: Ref2 Chap 3.2	Functions Modules and imports Students will create their own functions on a separate file and import them into their working environment
4	Introduction to Pandas Objects (Series, Dataframes, Index Object) Data Selection in Pandas TimeSeries Readings: Ref1 Part III.13, III.14, and III.15	Intro to Pandas Dataframes, Reading files, basic query
5	Adavanced Pandas Merge and Join Datasets Pivot Tables Aggregations and Grouping Eval and Query Readings: Ref1 Part III.19, III.20, III.21, III.23, and III.24	Advanced Pandas Groupby, join, data manipulation, and saving
6	Numpy Ufunctions; Broadcasting; Fancy Indexing Readings: Ref1 Part II.4, II.5, II.6, II.7, II.8, and II.10 | Ref2 Appendix A	Numpy Ufunctions and Broadcasting
7	Project Example with NetworkX, Seaborn, and Statsmodels Readings:	Expand your Stack e.g. Seaborn, matplotlib, statsmodels, networkX

Ref1: Vanderplas, Jacob T. "Python data science handbook: essential tools for working with data. 2nd Edition" (2023). (https://tinyurl.com/8j7atjx8)
Ref2: McKinney, Wes. Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. 3rd Edition" O'Reilly Media, Inc.", 2022. (https://wesmckinney.com/book/)

Demonstration of the Syllabus Coherence with the Curricular Unit's Learning Objectives

The learning units (LU) cover the learning outcomes (LO) as follows:
- LO 1 to 4 are addressed in LU0;
- LO 5 and 6 are addressed in LU1;
- LO 7 is addressed in LU2;

Teaching Methodologies

The curricular unit is based on a mix of theoretical and practical lessons with a robust and active learning component. During each session, students are exposed to new concepts and methodologies, case studies, and the resolution of examples. Active learning activities (debates, quizzes, compare-and-contrast) will place students at the center of the classroom, promoting peer learning and encouraging positive discussion. Computer activities will take place weekly during the practical lessons. Evaluation Elements:

EE1 - Participation in classroom activities (20%)
EE2 - Homework Assignments (40%)
EE3 – Final Exam (40%)

Schedule and Organization

The Programming for Data Science curricular unit consists of 7 weekly Lectures and Labs.
During Lectures, students will be introduced to the core theoretical concepts and examples.
In the Labs, students will be able to acquire hands-on practice solving exercises and challenges under the supervision of a teacher.
All lab activities will be conducted in the Jupyter Notebook Environment and using Python 3.X. For that reason, students must download the latest version of Anaconda (https://www.anaconda.com).

Grading

To successfully finish this curricular unit, students must score at least 9.5 points. The grading is divided into two seasons. Attendance in the second is optional for students who passed the curricular unit in the first season and can be used to improve their grades.

First Season

The first grading season is dedicated to continuous evaluation, which includes the following components:

- **Quizzes** (20% / 4 points) – It is a classroom activity. Weekly set of multiple-choice questions at the start of each Lecture. Quizzes will be performed on Socrative. Students can answer the quiz using their smartphones or computer laptops. Login details will be shared in Moodle during the first week of classes. Students are incentivized to debate with their colleagues during the quiz when possible. Out of the six quizzes, only the five best-scored quizzes will count toward the final grade.
- **Homework Assignments** (30% / 6 points) – Two problem sets designed to challenge students and incentivize their practice with programming. The assignments will be released at the end of weeks 3 and 6. You will have three days to submit the solution through Moodle. A penalty will be applied to late deliveries (1 point per day). The assignment is to

be delivered by pairs of students. However, the same pair of students can only submit one homework together. For instance, suppose that João and Pedro submit Homework 1 together. Thus, João and Pedro cannot submit Homework 2 together and must find an alternative.

- **Final Exam** (50% / 10 points) – Exam to take place during the 8th week of the semester. The exam consists of 40 multiple-choice questions covering fundamental programming concepts that are taught in both labs and lectures. The exam has a duration of 40 minutes, with correct answers awarding 0.5 points and incorrect answers deducting 0.2 points.

Second Season

The second grading season happens in January and consists of a multiple-choice exam with 40 questions. Correct answers count 0.5 points, and incorrect answers discount 0.2 points.

Demonstration of the coherence between the teaching methodologies and the learning outcomes

The presentation of theoretical concepts and methodologies, followed by application exercises, will provide students with the knowledge, skills, and abilities listed as learning objectives (LO).
Each evaluation element allows evaluating the LO listed, taking into consideration that:
- EE1 addresses LU0, LU1, and LU2;
- EE2 addresses LU0, LU1, and LU2;

Bibliography

- Lubanovic, Bill. Introducing Python: modern computing in simple packages. "O'Reilly Media, Inc.," 2014;
- VanderPlas, Jake. Python data science handbook: essential tools for working with data. "O'Reilly Media, Inc.," 2016.
- McKinney, Wes. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. "O'Reilly Media, Inc.," 2012.
- Grus, Joel. *Data science from scratch: first principles with Python*. "O'Reilly Media, Inc.," 2015
- Additional reading materials will be shared in Moodle with all the students, including documentation materials and book chapters;