BIO-645

Introduction to Applied Data Science (I2ADS)

Dayan Michaël Jérémy Pierre

Cursus	Sem.	Type
Neuroscience		Opt.

Language	English
Credits	2
Session	
Exam	Written
Workload	60h
Hours	**70**
Lecture	35
Exercises	35
Number of positions	

Frequency

Every year

Remark

Fall 2022

Summary

The "Introduction to Applied Data Science" (I2ADS) course is aimed at students of all levels to train them in the core computer science software stack and techniques forming the pillars of open & reproducible science. Registration in student portal & at https://tinyurl.com/i2ads2022 is compulsory.

Content

Schedule (all lectures are 2 hour-long)
=======

Lecture	Date & Time
Linux Part 1	TUE SEP 20, 2PM-4PM
Linux Part 2	MON SEP 26, 09AM-11AM
Linux Part 3	MON OCT 03 , 09AM-11AM
GIT Part 1	MON OCT 10 , 09AM-11AM
GIT Part 2	MON OCT 17, 09AM-11AM
GIT Part 3	MON OCT 24 , 09AM-11AM
GIT Part 4	MON OCT 31 , 09AM-11AM
Python Part 1	MON NOV 07 , 09AM-11AM
Python Part 2	MON NOV 14 , 09AM-11AM
Python Part 3	MON NOV 21 , 09AM-11AM
Python Part 4	MON NOV 28 , 09AM-11AM
Intro to Machine Learning Part 1	MON DEC 05 , 09AM-11AM
Intro to Machine Learning Part 2	MON DEC 12 , 09AM-11AM
Intro to Machine Learning Part 3	MON DEC 19 , 09AM-11AM

Syllabus
=======

o Linux
Part 1: Linux filesystem, Bash terminal and commands, Commands help / manual, Navigating the filesystem, Parsing files, Piping commands, File permissions, Super user privileges
Part 2: Writing bash shell scripts, Using a dedicated developing environment, Executing scripts / permissions, Bash variables, Strings, arrays, Control flow statements (for loop, if-else statements)
Part 3: Control flow statements [continued] (while loop, case/switch), Special script variables / exit codes, Arguments parsing with getopts, String/path manipulations, Hands on coding example of a full script

o Git
Git part 1: Git ecosystem, Git promotion model & typical workflow, Commit definition, Examining history, Hands-on: creating a history of commits on a local repo
Git part 2: Concept of Branches (as a pointer to a commit), Fast-forward merge, Introduction to Github, Remotes, Cloning, pushing and pulling, Tracking remote branches, Hands-on: creating a repo on Github and simulating team collaboration
Git part 3: Three-way merge, Tags, Merge conflicts, Hands-on: dealing with merge conflicts, Collaboration with fork / pull-request model, Hands-on: fork / PR model
Git part 4 Rebase, Cherry-picking, Finding a bug with bisection
o Python
Part 1: Introduction to Jupyter notebooks, Markdown, Variables and datatypes, Mutable vs Immutable datatypes, Strings and string manipulation, Lists, Dictionaries and tuples, Control structures: for loop, if-else statements, Constant interactive hands-on in between each concept
Part 2: Functions, None keyword, Function docstring, Type hinting, Development Environment (Visual Studio Code), Modules and imports, Debugging, Exceptions and asserts statements
Part 3: Object Oriented Programming Basics, Method attributes, Conda: creating, activating and configuring environments, Refactoring code, Creating modules and importing them in Jupyter notebooks
Part4: Numerical analysis and plotting with numpy: array creation, slicing and filtering, Plotting with matplotlib: line plot, scatter plot, bar plot, overlays, histograms, subplots, heatmaps

o Intro to Machine Learning (with Python)
Part 1: General usefulness of ML in science, Example of ML applied to simple problem of predicting projectile range, Choice of features and labels, Assessing model performance, MSE, Basics of optimization, parameter estimation, model fitting, Model generalizability, training/testing, Hands-on example with sklearn using linear (OLS) model, Bias-variance trade-off, Cross-validation, Hands-on example with sklearn using polynomial features and Pipeline objects
Part 2: Preventing overfitting with regularization, Hyper-parameter tuning, Nested cross-validation, Classification and classification metrics, SVM, Unbalanced classes, Dealing with regularization and class imbalance with SVM, Dealing with missing data / data imputation, Assessing model stability, Hyper parameter tuning with grid search
Part 3: Unsupervised techniques, Dimensionality reduction with PCA, Clustering with K-means, Assessing clustering performance

A step-by-step full example project should be implemented by the student at the end of the course (project content TBA).

Note

Attention: Registration in your student portal as well as at https://tinyurl.com/i2ads2022 is compulsory to be able to attend the course.
While the course will be made available in hybrid mode, both physically at Campus Biotech Geneva and remotely by connecting to our dedicated computing infrastructure, physical attendance is highly recommended. Credits will only be provided for those attending live at least 80% of the lectures (remotely or physically). An official EPFL email is required to attend the course.
Questions: Contact https://people.epfl.ch/michael.dayan
Learning outcomes: To implement in Python a data science project within a Linux environment while tracking code changes with the git version control system.

Keywords

Data Science; Linux; Git; Python; Machine Learning

Learning Prerequisites

Required courses

None

Recommended courses

None