Practical Data Science

Panos Louridas, Associate Professor, AUEB, louridas@aueb.gr

Overview

The course covers a large area of Data Science, focusing on practical applications with the Python programming language. Python is one of the most popular choices for handling big data. The clean syntax of the language, the availability of a large number of mathematical and scientific libraries, and a large programming community make it an ideal choice for tackling a large variety of real-world problems. As a result, Python is an essential component of a Data Scientist's tool chest.

The course treats tools, best practices, and practical applications of theoretical results in Data Science that enable us to process efficiently and leverage various forms of data.

Key Outcomes

By completing the course the students will be able to use Python in order to:

- Collect, process, and store, using appropriate tools and mechanisms, a variety of different data from the Internet.
- Carry out mathematical and scientific programming tasks using appropriate libraries.
- Avail themselves of tools to interact and process big data volumes.
- Employ Machine Learning methods and models.
- Carry out Natural Language Processing tasks.
- Solve a problem starting from a general statement, developing an algorithmic solution, then a fully-fledged implementation.
- Visualise their data and their results of their analyses.

Requirements and Prerequisites

This is a hands-on course. Students will spend a significant amount of time on writing programs and working with libraries and tools. We will use the Python programming language; although no previous knowledge of Python is assumed, it is assumed that students do have programming experience that they can use to reach proficiency in Python fast.

The course does not assume any prior experience in Python. However, basic knowledge of programming and computer science concepts is required.

Required Course Materials

There is no required textbook. All course materials will be provided in class and available for downloading.

Students will need to bring their laptops in class in order to try out interactively the material being presented.

Books

There are many books on the subject, and a lot of free resources on the Internet; the following selection provides a good foundation for those students who wish to delve deeper on the topics discussed in class:

- Michael Bowles, Machine Learning in Python: Essential Techniques for Predictive Analysis, Wiley, 2015.
- Joel Grus, Data Science from Scratch: First Principles with Python, O'Reilly, 2013.
- Wes McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, O'Reilly, 2012
- Ryan Mitchel, Web Scraping with Python: Collecting Data from the Modern Web, O'Reilly, 2015.
- Brett Slatkin , Effective Python: 59 Specific Ways to Write Better Python (Effective Software Development Series), Addison-Wesley, 2015.
- Cyrille Rossant, Learning Python for Interactive Computing and Data Visualization, 2nd ed., Packt Publishing, 2015.

Software/Computing requirements

Students will be able to run the examples and work with the material presented in the course using online tools and services. Instructions will be given at the relevant units during the course.

Students will also be able to run and work with most of the course material on their own computers. To to so, they should download and install the Anaconda Python distribution, available for all platforms at: https://www.continuum.io/downloads

We will be using Python 3.5 in the course, so make sure you pick that version on the download page. Instructions on how to use Anaconda will be provided in the course, but students should have downloaded and installed it in time for the first unit.

Grading

This is a practical course; students will be graded on their attendance and participation and on their competency to work with data in realistic problems in order to show practical results. This competency will be determined by course assignments.

Participation

In-class contribution is a significant part of your grade and an important part of our shared learning experience. Your active participation helps me to evaluate your overall performance. You can excel in this area if you come to class on time and contribute to the course by:

- Providing strong evidence of having thought through the material.
- Advancing the discussion by contributing insightful comments and questions.
- Listening attentively in class.
- Demonstrating interest in your peers' comments, questions, and presentations.
- Giving constructive feedback to your peers when appropriate.

Please arrive to class on time and stay to the end of the class period. Chronically arriving late or leaving class early is unprofessional and disruptive to the entire class. Repeated tardiness will have an impact on your grade.

Turn off all electronic devices prior to the start of class. Cell phones tablets and other electronic devices are a distraction to everyone.

Assignments

There will be three course assignments at two unit intervals.

1. The first assignment will be announced after the first two units and will count 20% towards the final grade.
2. The second assignment will be announced after the fourth unit and will count 20% towards the final grade.
3. The third assignment will be announced after the sixth unit and will count 20% towards the final grade.
4. The course project will be announced at the sixth unit; it will be a substantial undertaking in which the student will have to analyse a problem, find data to solve it, write the necessary programs, arrive at a conclusion, and fully document the solution and the results. The proposed course project will count 40% towards the final grade.

Late assignments will either not be accepted or will incur a grade penalty unless due to documented serious illness or family emergency. Exceptions to this policy for reasons of civic obligations will only be made available when the assignment cannot reasonably be completed prior to the due date, you make suitable arrangements, and give notice for late submission in advance.

Attendance Requirements

Class attendance is essential to success in this course and is part of your grade. An excused absence can only be granted in cases of serious illness or grave family emergencies and must be documented. Job interviews and incompatible travel plans are considered unexcused absences. Where possible, please notify the instructor in advance of an excused absence.

Students are responsible for keeping up with the course material, including lectures, from the first day of this class, forward. It is the student's obligation to bring oneself up to date on any missed coursework.

Code of Ethics

Students may not work together on graded assignments unless the instructor gives express permission.

Exercise integrity in all aspects of one's academic work including, but not limited to, the preparation and completion of all other course requirements by not engaging in any method or means that provides an unfair advantage. In any case of doubt, students must be able to prove that they are the sole authors of their work by demonstrating their knowledge to the instructor.

Clearly acknowledge the work and efforts of others when submitting written work as one's own. Ideas, data, direct quotations (which should be designated with quotation marks), paraphrasing, creative expression, or any other incorporation of the work of others should be fully referenced. No plagiarism of any sort will be tolerated. This includes any material found on the internet. Reuse of material found in question and answer forums, code repositories, other lecture sites, etc., is unacceptable. You may use online material to deepen your understanding of a concept, not for finding answers.

Please report observed violations of this policy. Any violations will incur a fail grade at the course and reporting to the senate for further disciplinary action.

Course Syllabus

The course comprises ten units of three hours each.

Unit 1: Introduction to Python

As the course assumes no prior knowledge of Python, we start with a rapid pace presentation of the main features of the language. In particular, we will overview syntax, data types, operators, control structures, functions, classes and objects, file handling. We will get a first view of the Python tools for Data Science (iPython, Numpy, Scipy, Matplotlib, Pandas, Scikit-learn) by walking through a typical example with real-world data.

Unit 2: Data Crawling

To carry out Data Science we need to crawl for data from various sources from the Internet. Popular web services like Facebook, LinkedIn, and Twitter, offer a wealth of data via specific Application Programming Interfaces (APIs). We will see how we can interact with these services and collect data, the most common data formats that we encounter in the Internet, and the ways we can read and parse them.

Unit 3: Data Storage and Retrieval; SQL and noSQL

We will juxtapose the two basic data storage technologies: relational (SQL) and non-relational (NoSQL) databases. Relational databases have been widely in use for several decades now, while non-relational databases have gained in popularity in the last few years. We will work with the MySQL relational database and the MongoDB NoSQL counterpart. We will examine how we can interact with them in Python. As each technology has its own strengths for specific kinds of data we will analyse their comparative advantages and the most appropriate application areas.

Unit 4: Numpy and Scipy

The Numpy and Scipy libraries provide the fundamental building blocks for performing mathematical and scientific computations in Python. They offer optimised data types and efficient algorithm implementations that we can either use directly or harness them to implement our own specialised solutions. We will overview their underlying principles and structures and we will investigate how we can draw on them to solve Data Science problems.

Unit 5: Pandas I

Often we need to examine data in different ways: filter them dynamically using various criteria, combine and merge data based on common elements, group data using specific values, etc. The Pandas library provides a rich set of such capabilities that enable us to manipulate, and work interactively with data at a high level of abstraction.

Unit 6: Pandas II

The Pandas library offers many facilities for working with time-varying data (time series analysis), with special applications to analysis of financial models. We will use it to become acquainted with the basic terms of time series processing and we will see how we can handle market data effectively.

Unit 7: Scikit-learn

Machine Learning (ML) is one of the most important branches of Data Science. It includes areas such as classification, regression, clustering, and dimensionality reduction, which are essential in processing data and solving many different problems. The Scikit-learn library offers a rich spectrum of Machine Learning functionalities and we will leverage it to approach some typical applications.

Unit 8: Natural Language Processing

The Natural Language Toolkit (NLTK) is a platform for building programs, in Python, to work with human language data. We will explore what natural language is about, applications of natural language processing, and what NLTK has to offer us.

Unit 9: Classification

Although there are many high quality tools, libraries, and frameworks in Python for Data Science there are always situations where we must develop a solution ourselves. To do that we must know how to proceed from a general problem description to its algorithmic solution, and then to an actual implementation. In this unit we will examine how we can solve the classification problem by using a classic algorithm for that task (ID3). We will start from the underlying principles of the algorithm and gradually develop a fully implemented solution.

Unit 10: Visualisation

To understand our data we need to be able to create graphical representations; visualisation is a rich and rewarding field. An image can be a thousand words, if done properly; or it can be a clutter of visual junk, or, worse, misleading, if we are not aware of some basic rules. After discussing the principles of visualisation we will show how the Matplotlib framework can help us create effective visualisations, both in screen and in print.