Computing for Data Science Course

Mark Fienup

Computer Science Department

University of Northern Iowa

Cedar Falls, Iowa 50614

mark.fienup@uni.edu

Abstract

This paper describes the Computing for Data Science course offered at the University of Northern Iowa (UNI) as part of the Data Science minor. This course acts as a bridge for non-Computer Science majors between UNI's CS1 course, Introduction to Computing, and our upper-level Computer Science course: Database Systems. The course's goals and organization are described including layout of topics, labs, and programming assignments.

1 Introduction

During the last couple years, Data Science programs in higher education have exploded. At the University of Northern Iowa (UNI) our initial response was to create an interdisciplinary Data Science minor using mostly existing courses, but with a few new courses tailored for the minor. This paper describes the Computing for Data Science course offered by the Computer Science department at the University of Northern Iowa (UNI) as part of the Data Science minor -- see Appendix A. This new course acts as a bridge for non-Computer Science majors between UNI's CS1 course, Introduction to Computing, and our upper-level Computer Science course: Database Systems.

The prerequisite course for Computing for Data Science is the Introduction to Computing (CS 1510). The Introduction to Computing course is taught using the Python programming language and expects no prior programming experience by students. Its course description is: "CS 1510. Introduction to Computing — 4 hrs. Introduction to software development through algorithmic problem solving and procedural abstraction. Programming in the small. Fundamental control structures, data modeling, and file processing. Significant emphasis on program design and style."

Traditionally, the prerequisites for Database Systems are both Data Structures and Discrete Structures where the Discrete Structures course is a discrete mathematics course taught by the Computer Science department to have a stronger CS focus. The Data Structures is also taught using Python and has a course description of "CS 1520. Data Structures — 4 hrs. Introduction to use and implementation of data structures such as sets, hash tables, stacks, trees, queues, heaps, and graphs. Additional topics include searching algorithms, sorting algorithms, and algorithmic time and space complexity analysis. Design and implementation of programs using functional decomposition."

The two high-level goals of the Computing for Data Science course are:

Goal 1. Prepare students to succeed in the Database Systems course by providing the crucial knowledge from the traditional prerequisite courses of Data Structures and Discrete Structures.

Goal 2. Advance students understanding and skills in Data Science.

2 Organization of the Course

Python is the programming language used in the course so we can build upon student programming skills learned in the Introduction to Computing. Plus, Python is frequently used by Data Scientists in the real world since many "third-party" Data Science tools/libraries (e.g., NumPy, Pandas, SciPy, TensorFlow, etc.) are available through Python. To avoid the complication of students installing these Data Science tools/libraries, the course uses Google Colab or "Colaboratory" [1] which allows access to all of these tools with a just a web-browser. No configuration is required and allows for easy sharing of files via Google drive and GitHub. In addition, it allows access to GPUs for computationally intensive Data Science tools free of charge.

My teaching philosophy is that students learn best if they are actively engaged [2]. Traditional lecturing to students is not very effective for student learning. Research has shown that 10 minutes is about the maximum attention span of students during lecture. However, this attention span clock can be reset by having students trying to apply what they just heard about. That is why I like to structure my classes into mini-lectures followed by small group activities -- typically a series of questions. After each small group activity, I like to discuss "correct" answers to the questions. To keep discussion focused and on track, I hand-out paper copies of the mini-lecture material (e.g., code, diagrams, timings, etc.) and corresponding questions with space for them to record their answers and the correct answers. Students find these material useful when studying for exams.

Since Computing for Data Science is intended to teach programming skills, I find "tightly coupled" laboratory activities useful for student learning. While the course does not have a separate formal "closed-laboratory" time, the course is taught on Tuesdays and Thursdays in an 90-minute periods. Typically, Tuesday's period introduces new material/topics in the mini-lecture fashion described above, and Thursday's period is a laboratory activity applied extensions to what was discussed in the previous Tuesday period. In laboratory activities are typically timing code, writing code segments, and answering related analytical questions.

The current textbook Foundational Python for Data Science [3] is more of a minimalistic Python review and Data Science reference.

2.1 Three Instructional Units

To achieve the two goals for the course, it is split into three roughly equal units:

Unit 1. Python Programming for Data Science: This unit mainly addresses Goal 1 by covering select topics from Data Structures and Discrete Structures courses. Details of unit 1 are listed in Table 1 below.

Unit 2. Data Science Libraries: This unit mainly addresses Goal 2 by covering select Data Science tools: NumPy arrays, SciPy library, Pandas Series and DataFrames, and data visualization using matplotlib, Seaborn, Plotly and Bokeh. Details of unit 2 are listed in Table 2 below.

Unit 3. Advanced Python and Data Science Tools: This unit mainly addresses Goal 2 since the "advanced" Python coverage is not directly needed by the Database Systems course. The "advanced" Python topics are OOP (object-oriented programming) and the `re` (regular expression) module. The OOP coverage introduces a new programming paradigm frequently used in Python programs which the students were not exposed to in their Introduction to Computing course. The advanced Data Science tools currently covered are the `nltk` (Natural Language Toolkit) package and a brief introduction to machine learning libraries: TensorFlow and Scikit-learn. Details of unit 3 are listed in Table 3 below.

Week Number	Tuesday Period Topics	Thursday Period Lab Activity
1	Introduction to Google Colab and signed integer representation of data	Python Review: arithmetic expressions and "control statements:" if-elif, while loops, for loops, nested loops
2	IEEE 754 floating point representation and review of functions in Python	Python Review: strings, lists, dictionaries, list-of-lists, dictionary of string keys with list values
3	Introduction of big-oh analysis and big-oh of list and dictionary methods; general idea of hashing	Practice determining big-oh notations and timings of lists and dictionary methods
4	Text-file usage in Python and .csv processing	Practice .csv processing and writing of output text-file
5	Sets in Python and their relationship to databases	Practice with Python sets and frozensets
6	Review for Test 1	Test 1

Table 1: Python Programming for Data Science Unit Details.

Week Number	Tuesday Period Topics	Thursday Period Lab Activity
7	NumPy scientific computing package	Practice with NumPy 1-D and 2-D arrays: views vs. copy, filtering values, and array methods
8	SciPy scientific computing package	Practice SciPy modules to do image processing and graph processing with NumPy arrays
9	Pandas Series and DataFrame data structures	Practice using Pandas by processing .csv data file
10	Visualization of data with matplotlib, Seaborn, Plotly and Bokeh	Practice data visualization
11	Review for Test 2	Test 2

Table 2: Data Science Libraries Unit Details.

Week Number	Tuesday Period Topics	Thursday Period Lab Activity
12	OOP programming in Python	Practice writing Python classes, inheritance and creating objects
13	Python `re` (regular expression) module	Practice using `re` module
14	Introduction to Machine Learning	Practice using Scikit-learn and NLTK text classifier
15	Work Day	Review for Final/Test 3

Table 3: Advanced Python and Data Science Tools Unit Details.

2.2 Programming Projects

In addition to the weekly laboratory assignments, larger programming projects are assigned to allow students to practice designing and writing larger programs in Python. Details of the programming projects for Spring 2023 are listed in Table 4 below.

Project Number	Brief Description of Programming Project	Desired Learning Outcome
1	Rock, Paper, Scissors program	Review of Python basics and practice functional-decomposition design and using functions
2	Math Tutor program	Practice functional-decomposition design, using functions, and user-input validation
3	JUMBLE puzzle solver	Practice functional-decomposition design, using functions, selection of efficient data structures, and text-files
4	Steganography – Embedding secret message into an image and decoding the message	Practice functional-decomposition design, using functions, SciPy modules and NumPy arrays to do image processing, and binary bit manipulation
5	Dice game program – (TBD)	Practice OOD and OOP in Python

Table 4: Programming Projects Details for Spring 2023.

3 Conclusions

The Computing for Data Science course is currently being offered for only the second time this Spring 2023 semester. Because the Data Science minor is relatively new at UNI and the Computing for Data Science is only taken by non-Computer Science majors, it only has six students enrolled for Spring 2023. All 6 students are juniors or seniors with a Mathematics-Statistics/Actuarial Science major, and all are doing well in the course.

The first offering of the Computing for Data Science course during the Spring 2022 semester had only seven students with only three of these students having a declared Data Science minor. The remaining four students were using this course as a substitution on their Interactive Digital Studies (IDS) major from the Department of Communication and Media department. While all of the students had taken the prerequisite Introduction to Computing course, a couple of the IDS majors struggled with the programming aspects of the course. However, one IDS major really liked the course and switched their major to Computer Science.

Hopefully, enrollment in the Computing for Data Science course will grow as we get more Data Science minors outside of Computer Science. Starting Fall 2023 UNI's new general education program, UNI Foundational Inquiry (UNIFI), will include a Data Science certificate which contains a non-major CS1 type course with a Data Science focus that can help populate the Data Minor with more non-CS majors.

References

[1] Google Colab URL: https://colab.research.google.com/

[2] J. Philip East, and Mark Fienup, "Questions to Enhance Active Learning in Computer Science Instruction," Proceedings of the 35th Annual Midwest Instruction and Computing Symposium, (CDROM) April 2002.

[3] Foundational Python for Data Science, Kennedy Behrman. Pearson Education. ISBN: 978-0-13-662435-6

Appendix A – Data Science Minor Requirements (2022 – 2023 University of Northern Iowa Catalog)

Data Science Minor

The Data Science minor is an interdisciplinary program that integrates computer programming, machine learning, statistics, predictive modeling and visualization to provide students with broad based skills for extracting gainful information from data that originate from a variety of sources. A final project (ideally with corporate or non-profit partnerships) will ensure that students employ their skills to solve a real-world problem.

Statistics:		
STAT 1772	Introduction to Statistical Methods	3
STAT 4784/5784	Introduction to Machine Learning	3
Computer Science:		
CS 1510	Introduction to Computing	4
CS 2150	Computing for Data Science	3-7
or		
CS 1520 & CS 1800	Data Structures and Discrete Structures	
CS 3140/5140	Database Systems	3
Physics:		
PHYSICS 4160/5160	Data Visualization, Modeling and Simulation	3
Required Data Science Project		2-3
CS 4800	Undergraduate Research in Computer Science	
or MATH 4990	Undergraduate Research in Mathematics	
or PHYSICS 3000	Undergraduate Research in Physics	
Total Hours		**21-26**