Course Syllabus

COSC 526/426 - Introduction to Data Mining, Spring 2023

University of Tennessee, Knoxville

Meeting Time and Place: Friday, 3:00 PM – 6:00 PM ET, Virtual

Course Credit Hours: 3 hours

Instructor: Dr. Michela Taufer

Dr. Taufer: *by appointment (with 24-hour notice) in zoom*

Teaching Assistants: Brandan Roachell

Office Hours: *by appointment (with 24-hour notice) in zoom*

Tool to ask questions and reach out to Instructor and TAs: COSC426-526-Spring2023 (]eecscosc42652-gln4300)

Invitations were sent to the student's UTK email address.

Course description

This course provides a practical introduction to data analytics, blending theory (e.g., of clustering algorithms and techniques for dealing with noisy data), and practice (e.g., using Apache Spark, Jupyter Notebooks, and GitHub). Over the semester, students will become familiar with modern data science methods, gain comfort with the tools of the trade, explore real-world data sets, and leverage the power of HPC and cloud resources to extract insights from data. Upon completing the course, students will have: used Jupyter notebooks to create reproducible, explanatory data science workflows; learned a modern MapReduce implementation, Apache Spark; implemented parallel clustering methods in Spark; studied strategies for overcoming the common imperfections in real-world datasets; and applied their new skills to extract insights from a high-dimensional medical dataset.

Course learning objectives

This course studies aspects of data analytics from a practical perspective. Specifically, during the semester, students will learn how to use distributed programming models such as MapReduce and how to implement clustering and classification algorithms in MapReduce to enable a scalable analysis of datasets across domains (e.g., medical, biological, ecoinformatics, and social sciences) on high-end clusters and supercomputers.

Targeted audience

The targeted audience of this course is engineering, computer science, and computational sciences students interested in data analytics and taking their first steps into the field. The course topics are relevant to these students because there is great interest in data analytics on HPC systems. The main challenge is a need for training courses that provide an easy-to-use interface to HPC systems and a software suite of scalable methods for data analytics. This course combines both aspects into a comprehensive semester-length course.

Student prerequisites and course requirements

Attendance is mandatory. **Students must bring their laptops to the lecture.** This advanced class presumes that you have **proficiency in coding with Python**. No book is required; reading material on the Internet and in ACM or IEEE digital libraries may be used.

Class environment

This section presents methods of instruction and the role of the students. At the beginning of the semester, each student will be given access to the tutorial GitHub repository with the tutorial materials (i.e., slides, assignments, and reading material). Each student will also be assigned a private repository for their course material (i.e., solutions to problems discussed in class and project material). Each student will have an account on an Apache Spark cluster.

The course is structured into several modules. Each module starts with a lecture in which the instructor presents a data-related topic or algorithm and assigns a suite of simple, practical, hands-on exercises that can be tackled and solved with the presented methodology. Students will review a proposed strategy for a solution, extend with their solutions, implement the solutions in the Jupyter Notebook (the entire hands-on exercise is annotated in the notebook, and dedicated coding cells are provided for the answers), and briefly discuss the findings with the instructor. This format requires active participation, critical thinking skills, and good programming skills in Python. During each module, students can work alone or in teams on the targeted, hands-on exercises; each student will be able to submit solutions to the assigned private repository before leaving the class and continue independently with any work that was not completed during the lecture during the rest of the week.

The initial module introduces the Jupyter notebook as an integrated development environment optimized to facilitate exploratory data analysis. Additionally, the version control system Git and its corresponding online community GitHub are introduced as integrated elements of the analytic workflow. In the subsequent modules, the instructor will teach the MapReduce programming model through Apache Spark, a parallelization framework that implements in-memory, fault-tolerant MapReduce abstractions. The instructor will then cover fundamental clustering algorithms and strategies for coping with missing or malformed data. Finally, the instructor will integrate all of these elements in the final module, enabling participants to perform exploratory data analyses of a nutritional/medical data set via Jupyter, parallelized with Spark, on a cluster or the Cloud. The tight feedback loop between implementation and visualization afforded by the Jupyter notebook facilitates rapid comprehension of module

contents. The Jupyter environment enables seamless visualization of results, thus building the students' intuition for how underlying algorithms and their parameters interact.

The final part of the semester is structured as a mini hackathon distributed across multiple lectures. During the mini hackathon, students will select a problem on big data and define a strategy to solve the problem supported by the tools learned during the first part of the semester. Examples of problems will be provided. Students are encouraged to bring their problems.

Detailed outline of the course

The course is structured into several modules:

Module I - Programming environment and infrastructure

Topics:

- Programming with Jupyter notebooks
- Version control with Git & GitHub

Practical hands-on exercise: Sequential text analysis and visualization using Jupyter notebooks

Module II - Introduction to MapReduce and Apache Spark

Topics:

- MapReduce as a programming model
- Map, Sort, Shuffle, Reduce workflow
- Partitioners and combiners
- Hadoop filesystem
- Overview of Apache Spark
- Resilient distributed datasets (RDDs)
- Parallel operations on RDDs
- In-memory computation in Spark

Practical hands-on exercise: Parallel text analysis (word, letter, and positional frequencies) using Spark

Module III - Clustering algorithms and their implementation in a MapReduce paradigm

Topics:

- Implementation of k-means from scratch in Spark
- Clustering with k-means and DBSCAN

Practical hands-on exercise: Application of parallel k-means from Apache Spark MLlib and parallel DBSCAN from the tutorial-provided library on real-world datasets

Module IV - Cope with missing data and applicability of MapReduce clustering techniques on real datasets

Topics:

- Taxonomy of missing data scenarios
- Procedures for handling missing data
- Parameter tuning for clustering algorithms

Practical hands-on exercise: Handle missing data and tune DBSCAN parameters for high-quality clustering of real-world datasets

Module V - Optimizing Apache Spark for HPC and Cloud platforms

Topics:

- Launching Apache Spark on a batch system (e.g., SLURM) or the Cloud (e.g., XSEDE Jetstream)
- Optimizing Spark's I/O for parallel file systems (e.g., Lustre)
- Managing RDD partitions

Practical hands-on exercise: Run previous exercises on HPC and Cloud resources with performance comparisons

Module VI – Working on real datasets

Topics:

- We are working with medical/dietary/social-economic data from National Health and Nutrition Examination Survey (NHANES).
- Working with Medicaid and Medicare data from a US state
- Work with Soil Moisture data from the ESA-CCI Initiative
- Work with Protein Diffraction images from Electron Laser (XFEL) beams
- Work on your dataset – need approval from the instructor

Practical hands-on exercise: Develop a set of critical questions and use the tools learned in this course to answer the questions; present the outcome in a poster and a 2-page extended abstract

How a student can be successful in this course

Each content module is paired with hands-on exercises that reinforce the concepts introduced and require proficiency in coding with Python. Over the course of the semester, each student will complete multiple hands-on activities. Each student must complete and submit his/her/their hands-on exercise in the assigned GitHub (private) repository.

To succeed, students shall:

- Attend the lectures and actively participate in the class activities
- Submit up to three questions/discussion points for the lecture breakout session before the following lecture (Friday before noon PM ET)
- Submit all the solution(s) to the weekly assignments in GitHub on time – Assignments are due weekly before the following lecture (Friday before 8 AM ET)
- Work on a project (projects require instructor approval) and implement the original solution(s) to the project problem(s)
- Submit a poster and a 2-page extended abstract describing the solution(s)
- Present the posters in a session scheduled for the last week of the semester.

Failing to succeed in one or more of the 5 points above will result in failing the course. This course does not have a final exam.

Mandatory rules associated with assignments:

- Assignments are due at 8 am ET on Friday after they are assigned
- To submit, push your Jupyter Notebook containing your solutions to your submission repository
- Your submission repository will have the same name as your GitHub username
- Please push all data and additional files that are distributed with the assignment. This is not required, but it helps the TAs in grading.
- You must push at least a partial submission by the due date
- If you do not make a partial submission, you fail the course
- If you make a partial submission, you will receive a temporary grade of "incomplete." You will then get one additional week to complete the assignment fully
- If your submission is incomplete or incorrect after the extra week, you will not automatically fail the course. However, it will negatively impact your final grade.
- It would be best if you pushed your solutions to your submission repository after completing each problem. This guarantees that you will get the extra week (unless you don't do the assignment at all)
- If you have extenuating circumstances (e.g., illness), go through the official University channels to report it, and you will be accommodated in -- "Absence Notifications" section of Hilltopics: https://hilltopics.utk.edu/academics/.

Academic integrity

Students may discuss hands-on exercises and the project with peers. However, all the work students submit **must be their own**, and all explanations must be in their own words. Students cannot write solutions for hands-on exercises in a group. Students cannot use the web to locate answers to any hands-on exercise. If students do not have time to complete an assignment, they should submit partial solutions than get answers from someone else. **Cheating students will be prosecuted according to university guidelines.** Students should familiarize themselves with their rights and responsibilities as explained in the Student Guide to University Policies (https://hilltopics.utk.edu/student-code-of-conduct/ (Links to an external site.))

Emergency absences

If serious illnesses, family emergencies, or other crises occur during the term, one of the key things that students must do is contact the dean of their college as soon as possible. This office can assist you in notifying faculty and validating what has happened for your teachers. Such validation will be necessary for you to make up missed class work (https://dos.utk.edu/absence-notifications/)

Disability services

Any student who feels s/he may need an accommodation based on the impact of a disability should contact Student Disability Services in Dunford Hall at 865-974-6087 or by video relay at 865-622-6566, to coordinate reasonable academic accommodations.