Syllabus

1. Course catalog description

Provides a practical introduction to big data mining and analytics, blending theory and practice. Over the course of 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 high performance and cloud resources to extract insights from data. Upon completing the course students learn

- How to create reproducible and explanatory data science outcomes.
- How to implement parallel clustering with Apache Spark.
- To overcome common imperfections in real-world datasets.
- To apply their new skills and extract insights from high-dimensional data.

2. Prerequisite

This course caters to those interested in data analytics and who have a basic understanding of programming concepts. All students are welcome, but knowledge of the Python programming language and basic statistical concepts will help you grasp theoretical concepts and practical applications covered in the course.

Prior experience with Jupyter Notebooks and Apache Spark is preferred but optional, as you will be able to bridge this knowledge gap independently using free resources like YouTube. Coursework provides accessible content to high performance computing systems and a software suite of scalable methods for data analytics.

Use this course to gain knowledge and experience in modern data mining and analytics techniques using real-world datasets. Outcomes include understanding supervised and unsupervised learning, data preprocessing, statistics, and deep hands-on coding in Python using Jupyter Notebook and parallel clustering with Apache Spark.

Topics include data preprocessing, unsupervised and supervised learning, Apache Spark, and data science interview preparation. Students build a strong foundation in data mining and its application to real-world scenarios by studying these techniques.

- Data preprocessing involves cleaning, transformation, and feature selection.
- Unsupervised learning covers clustering techniques.
- Supervised learning focuses on classification and regression.
- Apache Spark is a trusted open-source framework for parallel and distributed computing and big data analytics.
- The course also prepares students for data science interviews by providing tips, practice questions, and references for self-learning.

3. Course description

This course provides students with data mining and analytics essentials, including theoretical concepts and practical applications to succeed in intelligence activities. Students learn about supervised and unsupervised such as clustering, techniques for handling noisy data, and other modern data science methods. The course performs coding in Python using Jupyter Notebooks, providing hands-on experience while exploring real-world datasets.
Modern computing requires power and speed to emphasize high performance computing (HPC) by gaining experience implementing parallel clustering with Apache Spark. Students develop a solid understanding of the latest data mining blending theory and analytics techniques to address real-world dataset issues, such as missing data and outliers, statistical methods, and visualizations to extract data insights.

4. Topics

1. A practical introduction to data analytics using Apache Spark and Jupyter Notebooks.
2. Real-world datasets and strategies for common data imperfections.
3. High-performance computing and cloud resources for data mining and analytics.
4. Clustering and supervised algorithms for pattern analysis.
5. Data visualization techniques and data preprocessing methods.

5. Course structure

- ➔ Lectures for topic introduction, technique overview, and practical items.
  - ◆ Media includes audio, videos, and reading scientific articles.
- ➔ Assignments - perform practical problems in Jupyter Notebooks.
- ➔ Group discussion in breakout rooms.
- ➔ What if you need some more time to solve your problems?
  - ◆ Complete work and submit before the start of next week's class.

6. Books and resources

The following textbooks are specially mentioned for readings and learnings. There is a large universe of materials; bookmark O'Reilly Date topic.

A. Leskovec, J., Rajaraman, A., & Ullmane, J. D. (2020). *Mining of massive datasets* (3rd ed.). Cambridge University Press.

B. Vanderplas, Jake, 2016, Python data science handbook:, 1st ed.,O'Reilly, 2016
   a. https://jakevdp.github.io/PythonDataScienceHandbook/00.00-preface.html
   b. Chapters 2, 3, 4

C. Vanderplas, Jake, 2023, Python data science handbook:, 2nd edition O'Reilly, 2023

Additional core Python		Additional software training	
r.py.standard.library	Python documentation	cornell.into.to.python	cornell python training
PyPI · The Python Package Index	Python library index	pyu.PyMan.0.9.31	New York University Python training
Jupyter Community Forum	Search for tips and tricks	Get started with Jupyter Notebook	Notebooks training

7. Software and scientific installation

The Anaconda platform is highly engineered and automatically fixes many common Python installation issues. Select your operating system and run the defaults.

https://docs.anaconda.com/anaconda/install/windows
https://docs.anaconda.com/anaconda/install/mac-os/
https://docs.anaconda.com/anaconda/install/linux/

Anaconda3(64 bit) folder is visible in the start menu providing access to the Integrated Development Environment (IDE) Spyder, Jupyter Notebooks, and Anaconda Prompt (terminal). Spyder provides a console environment to code, view variables, and outcomes. While the classwork is organized in Notebooks the same work can be performed in Spyder. If curious, use Youtube et al. to self train.

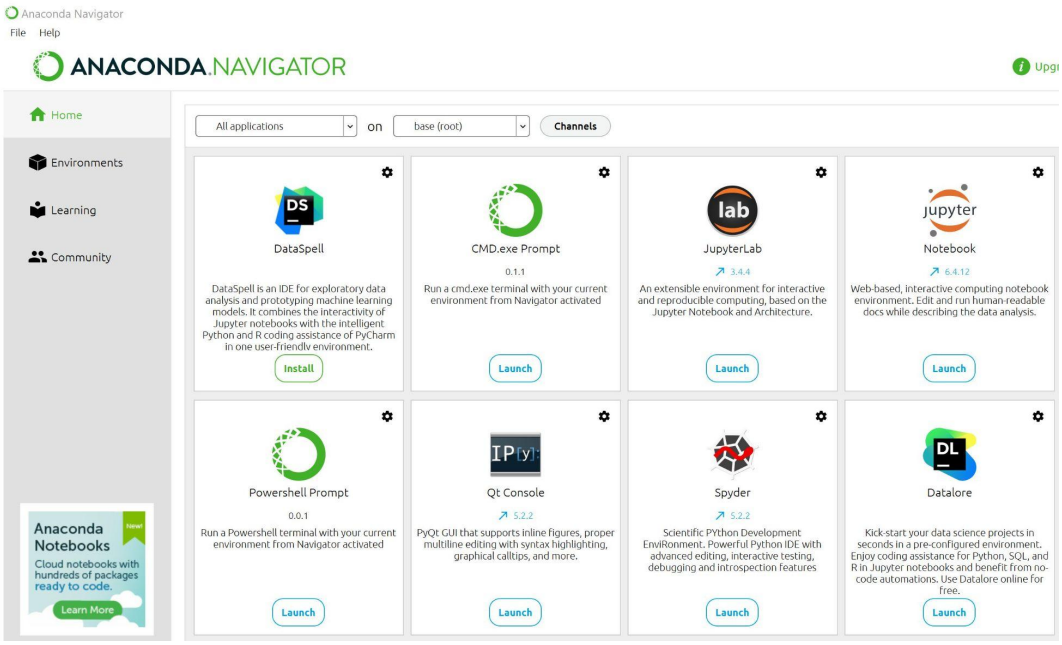
8. Jupyter Notebook

Your programming assignments will be done in Jupyter Notebooks. Jupyter Notebooks will allow you to create and share documents that contain live code, equations, visualizations and narrative text. Similar to a professional work environment, ensure to build familiarity with Github to source all your data data, class readings, notebooks, and syllabus.

Note: JupyterLab is a great alternative to Jupyter Notebook for portable code editing executed anywhere.

8.1 Launch Jupyter Notebook

From the Anaconda Navigator, select Jupyter Notebooks (not JupyterLab)
This starts a local web server and opens in your computer's default browser/



This interface will be used to:

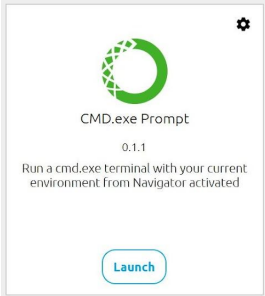
- create new Jupyter Notebooks
- upload and download notebooks for sharing
- upload and download data files for performing analysis within a Notebook

8.2 Validate required Python scientific libraries

Launch the Anaconda Navigator interface from the Start Menu and familiarize yourself with this interface. Inspect the "Environments" tab to view installed software packages and confirm the installation of Numpy, Pandas, Matplotlib, and PySpark. If missing, *CMD.exe Prompt* to open terminal and install.

- pip install pandas
- pip install numpy
- pip install matplotlib
- pip install pyspark

Need help?



<Shadow Box>

Resources [H3]

- Anaconda for windows
- Anaconda for mac-os
- Anaconda for Linux
- Install scientific packages.
- Reading: Software installation
- Reading: Using GitHub - class file exchange

Additional Resources

- Anaconda installation documentation.
- Jupyter Notebook documentation (including get started guides).
- Jupyter Discourse Forum.
    - Search here for tips, tricks, and solutions.
- Python Package Index (pypi)
- Spyder IDE - an alternative programming environment to Notebooks called an *integrated development environment* (IDE). Spyder is a sister environment to Notebooks providing an interactive console to view data, variables, and outcomes. It is not covered in the course but works alongside Jupyter Notebooks.
- GitHub - a place for storing files, searching for ideas, and framing interactive Jupyter Notebooks environments.
    - All data mining and machine learning scientists should have a page!

9. Class files

https://github.com/cosc-526/cosc.526.home.page/upload/main