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

Use the following textbooks for course readings and ongoing learning. Also in course Github: reference.resources.cosc.526

A. Leskovec, J., Rajaraman, A., & Ullmann, 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

D. Use the following table for additional Python reference and training

Additional core Python		Additional software training	
r.py.standard.library	Python documentation	cornell.intro.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.

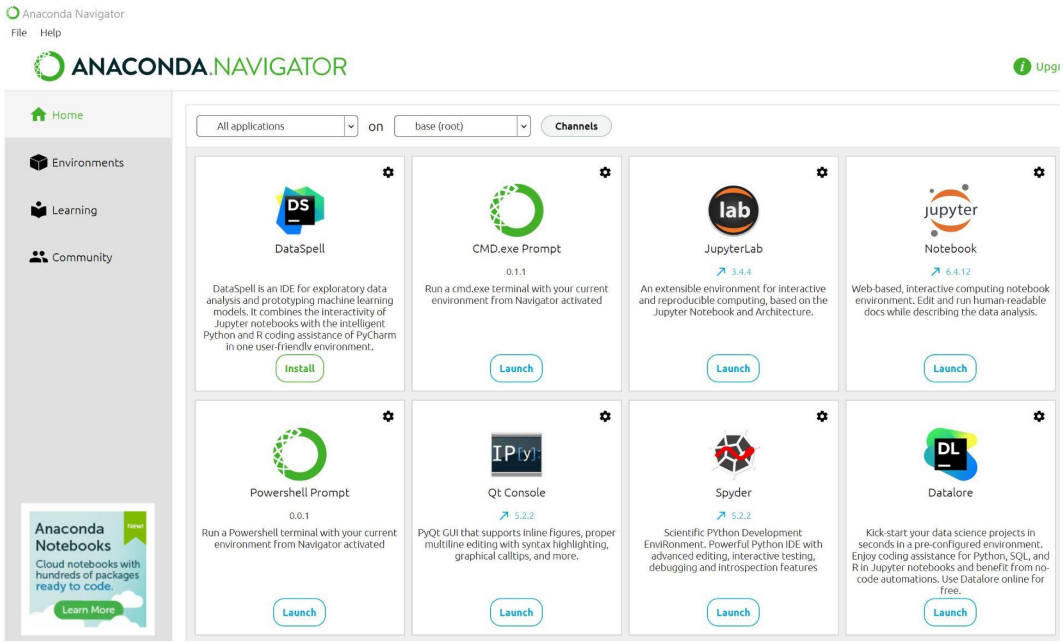
8. Jupyter Notebook

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/



Use the interface to:

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

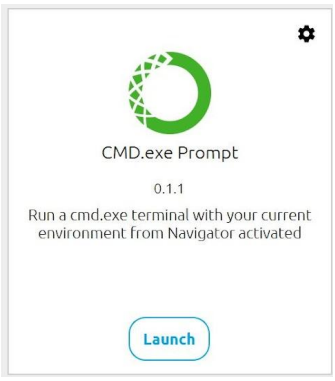
8.2 Validate required Python scientific libraries

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

Find, install and publish Python packages with the Python Package Index

- https://pypi.org/



9. Class repository of course files

Download class files including, amongst others, exercises, assignments, readings, and syllabus. Whether files are stored in GitHub, Google Drive, or any other university

storage location, it is the ***student's responsibility to know every single file*** in the repository. For instance, class discussions may lead to including new materials to support a new or changed activity or the addition of a scientific article to read for an online group discussion. Such items are deposited in the class repository and organized by name to help you quickly find a file.

- course GitHub
- course assignment and exercise Jupyter Notebook (.ipynb)
  - note: download .ipynb file from GitHub and upload to Jupyter Notebooks
- course assignment and exercise raw Python code (.py)

9.1 Examples of file name descriptions

➔ a. = article (news, journal) ➔ assign. = assignment	c. = cheatsheet code. = .py or .ipynb	g = graphic
howTo. = explanandum	py.M. exercise or assignment python file	r = reading

note: your UTK professor determines if a repository other than GitHub is used

File names	Purpose, description
https://github.com/cosc-526/cosc.526.home.page	Github source location of all files*
r.M.0.syllabus.cosc.526	reading.Module.0; course syllabus
howTo.Use.jupyter.Notebook.from.Bryn.Mawr.cololege.ipynb	howTo .ipynb file (Python Jupyter notebook) with details for formatting text and code Notebook code blocks by Bryn Mawr college
howTo.article.how.read.research.paper.pdf	howTo .pdf file of an "article" i.e. scientific journal article, on how to read scientific journal articles
Assignment name samples	
assign.M.1.assignment.1.covid.data.pdf	assignment for module.1,covid assignment
assign.M.2.assignment.2.pdf	assignment for module.2
d.M.1.10.assignment.covid.data.variants.csv	data for Module 1, subsection 10, covid assignment

note: your UTK professor determines if a repository other than GitHub is used

10. Topics by Module

A practical introduction to modern data analytics, blending theory such as clustering algorithms and noisy data techniques, and practice with tools like Apache Spark, Jupyter Notebooks, and GitHub. Real-world datasets facilitate insights and learning.

M	Module Topics	Practical Exercise
1	Introduction to Data Mining	Module 1 is a foundational introduction to data mining, providing an overview of its importance and applications for classification, regression, anomaly detection, and time series forecasting. Through real-world examples, discover how data mining can help answer complex questions and discern meaningful information. To achieve the course objectives, Module 1 provides a comprehensive overview of the Python coding environment necessary for assembling and implementing data mining techniques. By leveraging commonly used frameworks, you will develop the skills to navigate data structures and coding paradigms to appreciate the mechanics of data mining, machine learning, and artificial intelligence. Module 1 establishes the grounding and knowledge necessary to tackle more complex topics. Whether new to data mining, launching your graduate career, or deepening existing skills, please complete the readings, software installations, and data assignments to help ensure success and mastery of this powerful and pervasive technology.
2	Data Preprocessing	Module 2 on data processing is a comprehensive guide that equips students with data cleaning, integration, selection, imputation, reduction, transformation, discretization, outlier detection, and summarization skills. It begins by addressing common data quality issues and teaches programming methods using Python libraries pandas, NumPy, and scikit-learn. Practical exercises provide hands-on experience, while advanced topics cover data manipulation, filtering, grouping, merging, plotting, and visualization. The course emphasizes preparedness for on-the-job analysis challenges and learning new Python Package Index libraries (pypi)—relevant links: https://pandas.pydata.org/ and https://numpy.org/.
3	Algorithms: Unsupervised	Module 3 covers unsupervised learning algorithms, which identify patterns and relationships in data without a specific outcome variable in mind. Topics may include clustering, dimensionality reduction, and anomaly detection. Unsupervised algorithms: K-means clustering, hierarchical clustering, PCA, SVD, t-SNE, DBSCAN, Isolation Forest, LOF, and autoencoder. Optionally covered algorithms may include GMM, NMF, ICA, ARM, and LDA.
4	Algorithms: Supervised	Module 4 covers supervised learning algorithms used to predict an outcome variable based on a set of input variables. Topics may include regression, classification, and model evaluation. Supervised Algorithms: linear regression, logistic regression, decision trees, random forest, gradient boosting, naive Bayes, KNN, SVM, and neural networks such as MLP and CNN. Optionally covered algorithms may include Ridge/Lasso/Elastic Net regression, AdaBoost, XGBoost, LSTM, RNN, and deep learning architectures such as autoencoders and GANs.
M	Module Topics	Practical Exercise
5	Part I of III: Apache Spark	What is PySpark Spark Architecture and Cluster Setup PySpark Data Structures PySpark's Data Processing Operations Data Manipulation and Transformation

		Analysis and Visualization Assignment 5: Exploratory Analysis with PySpark RDD and DataFrames
6	Part III of III: Apache Spark	Spark SQL Machine Learning with PySpark Advanced PySpark PySpark in Production Assignment 6: PySpark SQL and Machine Learning
7	Part III of III: Apache Spark Review final project success requirements	Perceptrons and Linear Regression Logistic Regression and Naïve Bayes k-Nearest Neighbors and Decision Trees Random Forest and Support Vector Machines (SVM) Neural Networks and Unsupervised Learning Final Project requirements
8	Data science interviewing	Understanding the Interview Landscape Mastering Technical and Analytical Skills Effective Communication and Interview Strategies Case Studies and Real-World Challenges Assignment 8: Mock Interviews and Interview Preparation
9	Recommender Systems	Introduction to Recommender Systems: Data Collection and Preprocessing: Collaborative Filtering: Content-Based Filtering: Matrix Factorization Methods: Evaluation Metrics for Recommender Systems: Handling Cold Start and Diversity: Hybrid Recommender Systems: Assignment 9: Build a Recommender System
10	Final Project presentations	Knowledge overvew Staying Knowledgeable in an Evolving Field AI Trends and Challenges Professional Development; Thrive to Survive AI's Hardware Value of AI Research Generative AI: Purpose Goals and Outlook Deliver inclass presentations

11. Additional resources

Keep in mind your studying an artificial intelligence field growing in enormity. Here are a few suggestions to get connected with what's happening today

- identify industry leaders such as DeepLearning.AI's Andrew Ng, Microsoft researcher John Langford, and Google's Alekh Agarwal.

- follow machine learning and artificial intelligence information exchange on LinkedIn and Twitter to find new knowledge kernels and be on the cutting edge such as with ChatGPT engineering prompting.

11.1 Additional Python resources

- Anaconda for mac-os
- Anaconda for Linux
- Anaconda for windows
- Install scientific packages.
- Anaconda installation documentation.
- Anaconda for windows
- 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!

12. URL paths to miscellaneous links mentioned

- https://docs.jupyter.org/en/latest/start/index.html
- https://hunch.net/~jl/
- https://alekhagarwal.net/
- https://learn.deeplearning.ai/chatgpt-prompt-eng/