CMU ONLINE
Carnegie Mellon University

Syllabus Overview Sample: Foundations of Computational Data Science

Program: Machine Learning & Data Science Certificate Program

Faculty: Dr. Carolyn Rose, Professor of Language Technologies and Human- Computer Interaction, School of Computer Science

Course Number: 11973 **Units:** 12

Zoom Class: Weekday Evening, ET

Course Description

This course introduces foundational concepts, learning material, and projects related to the three core areas of Data Science: Computing Systems, Analytics, and Human-Centered Data Science. Students completing this class will be prepared for further graduate education in Data Science and/or Artificial Intelligence. Students acquire skills in solution design (e.g., architecture, framework APIs, cloud computing), analytic algorithms (e.g., classification, clustering, ranking, prediction), interactive analysis (Jupyter Notebook), applications to data science domains (e.g., Natural Language Processing, Computer Vision) and visualization techniques for data analysis, solution optimization, and performance measurement on real-world tasks.

Course Learning Objectives

By the end of this course, students will be able to:

- Understand the Data Science Process: From problem identification to ethical data solutions, students will learn to define analytic requirements and frame appropriate questions to guide the solution design process.
- Design and Implement Data Solutions: Develop a robust data-gathering plan that incorporates data governance principles, ensures data integrity, and safeguards data security.
- Analyze and Interpret Data: Apply statistical inference, hypothesis testing, and various data analysis techniques to identify trends, patterns, and outliers in large datasets.
- Develop Data Models and Algorithms: Build and evaluate predictive models using supervised and unsupervised learning techniques, understanding the bias-variance tradeoff and conceptual complexity in model building.

- Process and Analyze Text Data: Gain proficiency in Natural Language Processing (NLP) by exploring language representation, modeling, and implementing NLP tasks and applications.
- Implement and Evaluate Advanced Models: Learn to select appropriate models,interpret their results, and deploy them effectively while understanding the nuances of model selection and interpretability.
- Harness Deep Learning Techniques: Understand the differences between CPU and GPU computation and apply deep learning methods in areas like computer vision.
- Explore Advanced NLP Techniques: Delve into advanced NLP methodologies like transformers and BERT, gaining insights from cutting-edge research in the field.

Faculty Introduction



Dr. Carolyn Rose, Professor of Language Technologies and Human-Computer Interaction, has the privilege of directing the Teledia lab, a large, interdisciplinary lab involving PhD, Masters, and undergraduate students, staff, and affiliates researching interactive and explainable Sociotechnical Artificial Intelligence from a highly interdisciplinary perspective. From a machine learning perspective, we push the frontier of learnability and generalizabiliry through deeply data focused explorations of inductive biases. In particular, we develop and explore novel representations and architectural elements using a problem-driven approach motivated by error analysis and exploratory data analysis, with a current emphasis on abstraction and decomposition, which are arguably two of the greatest challenges for LLMs. We investigate these issues across multiple problem areas including multimodal conversational process analysis, multimodal document understanding, clinical text processing, knowledge based question answering, and language models of code.

Our work is particularly known for the way it bridges between deep, theoretical insights from theories of language and interaction on the one side (e.g., social psychology and cognitive psychology, sociolinguistics, and discourse analysis) and computational modeling technology on the other (e.g., deep learning, LLMs, neurosymbolic reasoning). The key enabler of effective machine learning is the measurement of capabilities, operationalized in computational objectives, and embodied in benchmarks. We have collaborated on the development of benchmarks and challenge data sets in coreference for dialogue, textual entailment, and event ordering, and are working on new benchmarks for code translation and code review.

Course Topics

Topic	Modules & Descriptions
Introduction to the Course, Setting Up Azure Account, Accepting Azure Course Subscription	• Define computational data science • Define big data • Demonstrate operations in a Jupyter Notebook • Set up an Azure account and an Azure subscription • Recite the workflow to submit coursework in a Jupyter notebook using a Bash submitter
Introduction to Computational Data Science, Problem Identification, and Ethical Data Solutions	• Evaluate the effectiveness of different data storage locations by comparing their speed and cost, applying the principle of locality within the context of memory hierarchy structures. • Understand the notions of and distinctions between abstract data types and (concrete) data structures. • Explain the costs of implementing the data types above with different underlying data structures, such as arrays, lists, trees or combinations of thereof. • Understand how time and memory resource costs of algorithms and data structures are estimated with asymptotic cost analysis.
Data Structures, Data Traversal Algorithms, Asymptotic Complexity	• Explain the impact of variable representation on memory usage and accuracy, and the costs of using different data structures (arrays, lists, trees) for implementation in Python. • Describe the functionality of key abstract data types (sequences, sets, tables, graphs) and their growth in runtime based on input size. • Match appropriate data structures to specific applications, considering algorithmic requirements and cost-efficiency.
Memory Allocation, NumPy, Pandas, Broadcasting	• Evaluate the efficiency of sparse matrices in space and computation optimization, and compare key formats (COO, DOK, CSR, CSC) for their practical strengths and limitations in data storage. • Illustrate the application of sparse matrices across various data science domains, assessing their advantages and disadvantages. • Apply NumPy arrays to manipulate numerical data efficiently, exploring vectorized operations, broadcasting, and aggregation functions to perform complex mathematical computations. • Utilize Pandas for data manipulation and analysis by employing its DataFrame and Series structures to filter, sort, group, and visualize large datasets effectively.
Linear Algebra, Vectorization, Recommender	• Define vectors and matrices, demonstrate their operations, and explain the use of special types of matrices in practical machine learning scenarios.

Systems	● Describe the benefits of vectorization in computational efficiency, identify conditions that favor its use, and convert iterative processes into vectorized implementations. ● Explain the business implications of recommender systems and the intuition behind collaborative filtering techniques including user-user, item-item, and matrix factorization, and their implementation in algorithms. ● Analyze data sparsity in recommender systems to select and apply the most appropriate sparse matrix format for efficient data representation and processing.
Data Visualization, Exploratory Data Analysis, Matplotlib, Seaborn	● Demonstrate the nine rules of graphical excellence in data visualization, rank various graphical representations by effectiveness in perceptual tasks, and apply visual vocabulary to create impactful data presentations. ● Identify and calculate summary statistics used in exploratory data analysis to describe and infer patterns or anomalies in data. ● Identify and customize components of a Matplotlib plot, arrange multiple plots in a single graph or grid, and integrate Matplotlib with Seaborn to enhance visualizations. ● Demonstrate the use of visual elements like color and point size to represent high-dimensional data in two-dimensional plots, and construct common plot types as per the visual vocabulary guidelines.
Text Data Processing, Regular Expression, Web Scraping, BeautifulSoup	● Apply regex syntax and grouping to identify complex patterns. ● Use NLTK for preprocessing text through stopword removal, tokenization, and POS-based lemmatization, and explain the construction and visualization of frequency-based text features. ● Construct and interpret GET requests to retrieve data from web APIs, and use BeautifulSoup to navigate and extract information from HTML documents. ● Describe statistical inference methods to estimate and make decisions about population parameters. ● Recite the fundamentals of statistical hypothesis testing in data science. ● Compare various techniques for representing domains as feature vectors, and explore how different featurizations affect the performance and task utility.
Machine Learning	● Analyze feature construction and cross-validation methods for improving model performance in linear regression, binary classification, and multiclass classification with sklearn. ● Evaluate and compare the performance of linear regression, logistic regression, and SVM models, using cross-validation and hyperparameter tuning. ● Design and implement a machine learning pipeline with feature construction, model training, hyperparameter tuning, and text

	data visualization using sklearn and yellowbrick.
Natural Language Processing, Classical Language Models	● Analyze feature construction and cross-validation methods for improving model performance in linear regression, binary classification, and multiclass classification with sklearn. ● Evaluate and compare the performance of linear regression, logistic regression, and SVM models, using cross-validation and hyperparameter tuning. ● Design and implement a machine learning pipeline with feature construction, model training, hyperparameter tuning, and text data visualization using sklearn and yellowbrick.
Model Evaluation, Model Deployment	● Understand the process of transforming and normalizing tensor data and explain its significance in preparing data for computer vision models. ● Apply PyTorch to implement Logistic Regression, LeNet, and AlexNet models, demonstrating the ability to build and train these neural networks for computer vision tasks. ● Evaluate the trade-offs between training time and cost when using CPU vs GPU.
Attention, Transformers, LLMs (Large Language Models), BERT/Autoregressive	● Create a fine-tuned computer vision model and deploy it to a public API endpoint using Azure, demonstrating the ability to manage the entire workflow from model training to deployment.
BERT, Question-Answering, Prompting/Few-Shot Learning, Fine-Tuning	● Analyze the importance of context in word embeddings and evaluate the differences between traditional embedding techniques (e.g., bag-of-words, TF-IDF) and advanced models like Word2Vec and GloVe ● Apply advanced NLP models such as BERT, focusing on the implementation of attention mechanisms to enhance sequence-to-sequence tasks and other NLP applications ● Create custom NLP pipelines using transformers, incorporating multi-head attention mechanisms to improve model performance ● Understand the concept of language modeling and explain the principles behind generative models like GPT, focusing on their ability to generate text

NOTE: This is a sample designed to provide an overview of topics covered in the course based on previous course syllabi. All students will be provided an updated and final syllabus at the start of the course. Faculty, content, and schedule are all subject to change.