Data Mining in Engineering-CHE1147
Course Syllabus-2022

Nikolaos Anesiadis, PhD

1 Course Description

With continuous advances in processing power, storage capacity, sensors, and connectivity of electronic devices, enormous amounts of data are generated by different industries, businesses, tech companies and governments. Extracting useful knowledge from data requires interdisciplinary skills in scientific computing methods and algorithms. The broader term that captures all the skills mentioned is **data science**. Data-driven organizations leverage their data effectively and generate business insights that enable better decision-making.

In this course, we will examine how data science can be used to improve decision-making. We will present both the theoretical background and practical application of data science. Students will gain hands-on experience on major data science techniques and tools and how they are applied to real-world data sets.

A data science project typically consists of five steps: problem definition, data preparation and pre-processing, modeling, evaluation and presentation/communication.

The **first** requirement to do data science is **domain knowledge**. Defining an engineering, scientific or business problem may be obvious and clear or challenging and complex. Domain knowledge is something that each individual brings in the class, since you are the expert of your research area, and it will help you to clearly define a research problem. Most academic courses in statistics, machine learning and data mining often present students with clearly defined problems ready to pass to algorithmic solutions. In reality, defining a problem is an iterative process. In the first four weeks of the course, we will focus on defining the project problem you will work on throughout the semester.

The two skills that we will spend most of the time in this course are: **programming** and **applied math/machine learning**. These skills enable us to do **data engineering** and **data science**. The former involves the collection, storage, processing and transformation of data; the latter involves the application of algorithmic/mathematical computations and visualization techniques to answer the scientific problem in question. Also, some concepts of database systems are discussed at an introductory level.

2 Prerequisites

Intellectual curiosity, self-motivation and discipline are expected in every graduate course. The nature of the course does not allow anyone to rely on one textbook. Consulting several books and online sources (e.g. Stackflow, Github) is highly recommended. My personal motto is *"if I have a question, somebody else most likely has already asked it"*; it is almost always true.

As an introductory course in data science, we assume that students have **some basic** background knowledge of the following concepts:

- **Programming:** basic programming in Python, R or Matlab. A short introduction to Python will be given, but if you are comfortable in any other language you can use it. **IMPORTANT:** A laptop with a Python or R integrated environment installed is necessary to maximize the hands-on experience in class. Install one of the IDEs in section 7.1 on your laptop prior to the first class.
- **Linear algebra:** vectors, matrices, vector spaces, basis, matrix inversion, linear equations, optimization.
- **Statistics:** expected value, probabilities, distributions, hypothesis testing, ANOVA.
- **Algorithms:** be able to understand algorithms and logic written in pseudo-code.

3 Course Outline

Module #1: Introduction to Python and basic libraries: pandas, numpy, matplotlib. Data loading and wrangling: types of data, indexing and slicing, subselecting, filtering, merging and joining. Data pre-processing (imputation, scaling, etc.), Data aggregation and group operations.
Source: Chapter 4-9 Python for Data Analysis

Module #2: Feature engineering: numerical and categorical variables, feature selection Visualization and exploratory data analysis (EDA): line, scatter, box, hist, pdf, cdf, hexbin plots.
Source: Chapter 2, 5 from Feature Engineering for Machine Learning, paper by Guyon and Elisseef (2003)

Module #3: From a scientific or a business problem to a data science workflow. Problem formulation. Decision making. Definition of the modeling universe. Supervised vs unsupervised learning. Classification and regression. Overfitting. Bias-variance. Outputs: probabilities-classes, numeric estimates-range.
Source: Chapter 2-5, 7, 8 and 12 from Data Science for Business

Module #4: Unsupervised ML algorithms #1: Preprocessing and scaling. Projection methods, dimensionality reduction, feature extraction, manifold learning. Principal component analysis (PCA), Incremental and randomized PCA, Non-Negative Matrix Factorization (NNMF).
Source: Chapter 3 from Introduction to Machine Learning with Python, Chapter 8 from Hands-on Machine Learning with Scikit-Learn and TensorFlow, Chapter 14 from The Elements of Statistical Learning, Chapters 1-3 from Multi- and Megavariate Data Analysis

Module #5: Unsupervised ML algorithms #2: Clustering methods. k-means, agglomerative, DBSCAN.
Source: Chapter 3 from Introduction to Machine Learning with Python, Section 14.3 from The Elements of Statistical Learning, Section 10.3 from Introduction to Statistical Learning with R

Module #6: Supervised ML algorithms #1: Classification: logistic regression, k-nearest neighbors, support vector machines (SVM). Limitations and extensions. Kernels.
Source: Chapter 2 from Introduction to Machine Learning with Python, Section 4.4, 13.2, 13.3 and Chapter 12 from The Elements of Statistical Learning, Section 4.3 and Chapter 9 from Introduction to Statistical Learning with R, Chapter 4, 5 from Hands-on Machine Learning with Scikit-Learn and TensorFlow

Module #7: Supervised ML algorithms #2: Classification: decision trees, ensemble learning and random forests. Fine-tuning of trees, ensembles of trees and robustness. Feature importance.
Source: Chapter 2 from Introduction to Machine Learning with Python, Chapter 9, 10 from The Elements of Statistical Learning, Chapter 8 from Introduction to Statistical Learning with R, Chapter 6, 7 from Hands-on Machine Learning with Scikit-Learn and TensorFlow

Module #8: Model evaluation: cross-validation, grid search, overfitting, evaluation metrics (lift, ROC/AUC curves, confusion matrix, profiling), scoring. Most of the terms mentioned here are explained along with the respective algorithms in previous weeks. Here, we recap with the final project presentation in mind.
Source: Chapter 5 from Introduction to Machine Learning with Python

4 Grading

Evaluation will be based on 6 individual assignments that will be spread out throughout the semester. The assignments are part of the ongoing learning experience. They are designed so that you continue to learn and practice coding and modeling methods.

Assignment	Weight (%)	Deadline
Basic Python	10	Sep-28
Aggregations and for loops	10	Oct-12
Feature engineering	25	Oct-26
Unsupervised methods: PCA/PLS	20	Nov-9
Supervised learning methods	20	Nov-23
Cross-validation, grid-search	15	Dec-7

Late Submission: 10% of the marks will be deducted for each day late, up to a maximum of 3 days. After that, no submissions will be accepted.
Posting online: please do **not** post any of the notes or the assignments online.

5 Office hours

- **Please** use personal email **nikanesiad@gmail.com** for faster responses. Most questions will be answered through email and the discussion board on Quercus.
- Teaching assistants: TBD

6 Textbooks

6.1 Required

- Python for Data Analysis (Wes McKinney): a book by the creator of Pandas (code on Numpy and Pandas freely available); will be used throughout the course.
- Introduction to Machine Learning with Python (Sarah Guido and Andreas Muller): the code is freely available and will be used in this course
- Hands-On Machine Learning with Scikit-Learn and TensorFlow (Aurelie Geron)

6.2 Available for free

- Think Stats (Allen B. Downey): a book for exploratory data analysis in Python
- Think Python (Allen B. Downey): a book on programming on Python
- The Elements of Statistical Learning (Hastie, Friedman, and Tibshirani): an introductory book in machine learning with applications in R
- An Introduction to Statistical Learning (James, Witten, Hastie, and Tibshirani)

- Understanding Machine Learning: From Theory to Algorithms (Shalev-Shwartz and Ben-David)

6.3 Highly recommended

- Feature Engineering for Machine Learning (Alice Zheng and Amanda Casar): a book on an often neglected topic in data science.
- Data Science for Business (Foster Provost and Tom Fawcett): a resource with great insights and practical knowledge on data science and machine learning, minimum mathematics and programming.
- Multi- and Megavariate Data Analysis (MKS Umetrics AB): a great book on PCA-PLS by the creators of SIMCA, a commonly used software in bio and chemical industries for outlier and trends detection.
- Doing Data Science (Cathy O'Neil and Rachel Schutt): a little broader scope; examples in R.
- Pattern Recognition and Machine Learning (Christopher Bishop): a classic textbook with detailed mathematical derivations and explanations.

7 Resources

7.1 IDEs for Data Science

The use of Python or R is highly recommended. I will mostly be using Python through Anaconda and Jupyter Notebooks. The use of GUI-based software such as Weka, Orange, Rapidminer and KNIME is not recommended due to lack of flexibility (it will also prevent you from learning programming, data munging and algorithms).

- **Anaconda:** a Python scientific distribution, with Jupyter, JupyterLab and Spyder, a MATLAB-like development environment https://www.anaconda.com/download
- **Jupyter:** a web application interaction environment (also part of Anaconda) http://jupyter.org/install.html
- **PyCharm:** this is ideal for you if you are used to other JetBrain IDEs for JavaScript, HTML/CSS https://www.jetbrains.com/pycharm/download
- **RStudio** this is an open source professional software for R https://www.rstudio.com/products/rstudio/download3/

7.2 Python code: Notebooks and repositories in github

- Supporting code for Think Stats, 2nd Edition
  https://github.com/AllenDowney/ThinkStats2
- Code examples and exercise solutions from Think Python
  https://github.com/AllenDowney/ThinkPython
- Materials and IPython notebooks for "Python for Data Analysis" by Wes McKinney, published by O'Reilly Media https://github.com/wesm/pydata-book
- Code from "Feature Engineering for Machine Learning" by Alice Zheng and Amanda Casar https://github.com/alicezheng/feature-engineering-book
- Notebooks and code for the book 'Introduction to Machine Learning with Python' by Sarah Guido and Andreas Muller
  `https://github.com/amueller/introduction_to_ml_with_python`

7.3 Groups, email lists, miscellaneous sources

- O'Reilly newsletter: a publisher with a strong focus on programming and data science books
- KDNuggets: groups on Linkedin, Facebook, Twitter, personal website, email list
- Medium: follow several great data scientists

8 Do you really want to do Data Science?

Think about why you want to take this course. Data science is associated with taking very important decisions. A company's strategy or a scientific conclusion depends on your analysis. Ask yourself the following questions (from Eric Weber, a data-scientist), think carefully and if you answer 'Yes' or 'Most likely' to most of them, then you are in the right place. Also, the questions capture accurately the life of a data scientist.

1. Do I like working with ill-formed problems that do not have a clear problem formulation and solution? Dirty data sets from multiple sources?
2. Can I deal with ambiguity and frustration with sometimes difficult tasks?
3. Am I comfortable communicating with non-technical audiences?
4. Am I tenacious? Do I keep working on a problem until I deliver what was asked?

5. Am I interested in continually re-inventing and developing my skill set?
6. Am I okay having major responsibility for company decisions?
7. Am I willing to admit when I don't know something?