Frequentist Machine Learning
Fall 2020 Syllabus

Course Description: Statistical inference, exploratory data analysis. Regression methods such as Ridge, LASSO, Elastic Net. Classification methods such as Logistic Regression, SVM. Regularization and feature selection methods. Classification and Regression Trees including random forests and extreme gradient boosting. Model selection and cross validation. Data Visualization.

Textbooks:

http://statweb.stanford.edu/~tibs/ElemStatLearn/

Tentative sections to be covered: 3.1-3.6, 4.1-4.4, 7.1-7.7. 8.1-8.6, 9.1-9.2, 10.1-10.10,12.2-12.3. 14.3-14.6. 15.1-15.3

Other Books:

http://greenteapress.com/thinkstats/

www.amazon.com/Doing-Data-Science-Straight-Frontline/dp/1449358659

Data resources:

https://www.kaggle.com/datasets

http://archive.ics.uci.edu/ml/

http://www.kdd.org/kdd-cup

https://docs.google.com/spreadsheets/d/1JYAjpmAC_qosVqJH5vHrRopmxjNUPmKrUJbRUiSaJnY/edit#gid=0

Mini projects, details will be posted as they are assigned.

Linear Regression
Linear Classification
Cross-Validation
Random Forests
Xtreme Gradient boosted trees
Non-Negative Matrix Factorization
Market Basket Analysis

Grading: This course will be entirely project and participation based. There are 7 base projects, worth 10 points each. Complete all 7 projects perfectly and you get a C. There will be additional ways to earn points, such as reading papers, participating in paper discussions, and completing stretch goals on the projects. It works out like this:

7 projects, 10 points each = 70 total points
5 paper, 1 point for completing a paper evaluation, 1 point for participating in group discussion of paper, = 10 total points
Miscellaneous stretch goals = 20 additional available points.

Grading Scale:

90-100 points: A
80-89 points: B
70-79 points: C
60-69 points: D

Grading Rubric for MiniProjects:

7 points: Technical correctness of project
2 points: Quality of code (well organized, well commented, etc)
1 point: Summary of results. What did you learn in this project? Why did the algorithm perform the way it did? Etc.

Anti-Plagiarism method

Unfortunately, coding exercises are easily plagiarised. It is normal and productive to share snippets of code with colleagues, or to find code on the internet to use. It is not acceptable to turn in code you do not understand. In industry, a common practice is code review, in which before your code gets accepted into a larger code base, you sit with another coder and explain what your code does and why. We will employ this method to reduce plagiarism in this course. You and your group will need to explain each project to me, over zoom/teams. Time during class will be allocated for this, with sign up slots.

The above anti-plagiarism method is in addition to the policies found here:
https://cooper.edu/engineering/curriculum/academic-standards-regulations