**UCI-Math10**

**This is the repository for Math 10 Intro to Programming for Data Science**

**Math 10** is the first dedicated programming class in the Data Science specialization designed mainly for Math majors at University of California Irvine. Some of current de facto algorithms will be featured, and some theorems in Mathematics behind in data science/machine learning are to be verified using Python, and the format can be adapted to other popular languages like R and Julia.

**Prerequisites:**

**MATH 2D** Multivariate Calculus

**MATH 3A** Linear Algebra(can be taken concurrently)

**MATH 9** Introduction to Programming for Numerical Analysis

**Recommended:**

**MATH 130A** Probability I

**ICS 31** Introduction to Programming

**Lecture notes (Jupyter notebooks) are available in the Lectures folder.**

| **Lecture** | **Contents** |
| --- | --- |
| Lecture 1 | [Intro to Jupyter notebooks, expressions, operations, variables](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-01-introduction.ipynb) |
| Lecture 2 | [Defining your own functions, types (float, bool, int), Lists, IF-ELSE](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-02-Variables-and-Functions.ipynb) |
| Lecture 3 | [Numpy arrays I, tuples, slicing](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-03-Numpy.ipynb) |
| Lecture 4 | [Numpy arrays II, WHILE and FOR loops vs vectorization](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-04-Loops-vs-Vectorization.ipynb) |
| Lecture 5 | [Numpy arrays III, advanced slicing; Matplotlib I, pyplot](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-05-Matplotlib.ipynb) |
| Lecture 6 | [Numpy arrays IV, Linear algebra routines](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-06-Linear-Algebra.ipynb) |
| Lecture 7 | [Matplotlib II, histograms](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-07-Histogram.ipynb) |
| Lecture 8 | [Randomness I; Matplotlib III, scatter plot](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-08-Randomness.ipynb) |
| Lecture 9 | [Randomness II, descriptive statistics, sampling data](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-09-Statistics.ipynb) |
| Lecture 10 | [Randomness III, random walks, Law of large numbers](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-10-Random-Walks.ipynb) |
| Lecture 11 | [Introduction to class and methods, object-oriented programming](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-11-Class.ipynb) |
| Lecture 12 | [Optimization I: Optimizing functions, gradient descent](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-12-Gradient-Descent.ipynb) |
| Lecture 13 | [Fitting data I: Linear model, regression, least-square](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-13-Linear-Regression.ipynb) |
| Lecture 14 | [Optimization II: Solving linear regression by gradient descent](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-14-Linear-Regression-II.ipynb) |
| Lecture 15 | [Fitting data II: Overfitting, interpolation, multivariate linear regression](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-15-Overfitting.ipynb) |
| Lecture 16 | [Classification I: Bayesian classification, supervised learning models](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-16-Classification-I.ipynb) |
| Lecture 17 | [Classification II: Logistic regression, binary classifier](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-17-Classification-II-Binary.ipynb) |
| Lecture 18 | [Classification III: Softmax regression, multiclass classifier](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-18-Classification-III-Multiclass.ipynb) |
| Lecture 19 | [Optimization III: Stochastic gradient descent](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-19-Stochastic-Gradient-Descent.ipynb) |
| Lecture 20 | [Classification IV: K-nearest neighbor](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-20-K-Nearest-Neighbor.ipynb) |
| Lecture 21 | [Dimension reduction: Singular Value Decomposition (SVD), Principal Component Analysis (PCA)](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-21-Principal-Component-Analysis.ipynb) |
| Lecture 22 | [Feedforward Neural Networks I: models, activation functions, regularizations](https://github.com/scaomath/UCI-Math10/blob/master/Lectures/Lecture-22-Neural-Network-I.ipynb) |
| Lecture 23 | [Feedforward Neural Networks II: backpropagation](https://github.com/scaomath/UCI-Math10/blob/master/Lectures/Lecture-23-Neural-Network-II.ipynb) |
| Lecture 24 | [KFold, PyTorch, Autograd, and other tools to look at](https://nbviewer.jupyter.org/github/scaomath/UCI-Math10/blob/master/Lectures/Lecture-24-Advanced-Tricks.ipynb) |

**Labs and Homework**

There are two Labs per week. One is a Lab exercise, aiming to review and sharpen your programming skills. The other is a graded Lab assignment, which is like a collaborative programming quiz. Homework is assigned on a weekly basis, the later ones may look a mini project. Lab assignments' and Homework solutions are available on [Canvas](https://canvas.eee.uci.edu/).

**Textbook**

No official textbook but we will use the following as references: [Scientific Computation: Python Hacking for Math Junkies. Version3, With iPython](https://github.com/biomathman/python-book/) (Math 9 reference book)

[Python Data Science Handbook. Online version](https://jakevdp.github.io/PythonDataScienceHandbook/)

**Software**

Python 3 and Jupyter notebook (iPython). Please install [Anaconda](https://www.anaconda.com/download). To start Jupyter notebook, you can either use the Anaconda Navigator GUI, or start Terminal on Mac OS/Linux, Anaconda prompt on Windows: in the directory of .ipynbfile, run the command jupyter notebook to start a notebook in your browser (Chrome recommended). If Jupyter complains that a specific package is missing when you run your notebook, then return to the command line, execute conda install <name of package>, and re-run the notebook cell.

**Final Project**

There is one final project using Kaggle in-class competition. A standard classification problem similar to the Kaggle famous starter competition [Digit Recognizer](https://www.kaggle.com/c/digit-recognizer) based on MNIST dataset will be featured. You will use the techniques learned in class and not in class (e.g., random forest, gradient boosting, etc) to classify objects.

* [Winter 2019 final project: Learn the handwritten characters in ancient Japanese](https://www.kaggle.com/c/uci-math-10-winter2019)
* [Spring 2019 final project: Is your algorithm fashionable enough to classify sneakers?](https://www.kaggle.com/c/uci-math10-spring2019)
* [Winter 2020 final project](https://www.kaggle.com/c/uci-math-10-winter2020)

**Acknowledgements**

A major portion of the first half of the course is adapted from [Umut Isik's Math 9 in Winter 2017](https://www.math.uci.edu/~isik/teaching/17W_MATH9/index.html) with much more emphases on vectorization, and instead the materials are presented using classic toy examples in data science (Iris, wine quality, Boston housing prices, MNIST, etc). Part of the second half of this course (regressions, classifications, multi-layer neural net, PCA) is adapted from [Stanford Deep Learning Tutorial](http://ufldl.stanford.edu/tutorial/)'s MATLAB codes to vectorized implementations in numpy from scratch, together with their scikit-learn's counterparts.