PURDUE
KRANNERT
SCHOOL OF MANAGEMENT

# MGMT 590: Computing for Analytics

Summer 2017


| | |
|---|---|
| Professor: | Yaroslav Rosokha |
| E-mail: | yrosokha@purdue.edu |
| Office Hours: | 4.15PM – 5.15PM |
| Office: | Krannert 410 |
| Phone: | 765-496-3668 |
| TAs: | TBA |
| TA Office Hrs: | TBA |


## I. Class Description

The main goal of this course is to introduce students to the tools and methods for data analytics. The course complements other courses in BAIM program with a bottom-up, programmatic approach of how to retrieve, manipulate, visualize, and analyze the data. The course will focus on challenges associated with large datasets and how algorithms and data structures can aid in resolving some of those challenges. The course will introduce relevant programming techniques in Python.

## II. Tentative Schedule

| Date | Time | Class | Contents | Due |
|---|---|---|---|---|
| 7/3 | 1 pm | 1 | Introduction | HW0 |
| 7/3 | 4.15 pm | 2 | Data Structures | |
| 7/5 | 1 pm | 3 | Computing Summary Statistics | HW1 |
| 7/5 | 4.15 pm | 4 | Data Visualization I | |
| 7/7 | 1 pm | 5 | Data Visualization II | HW2 |
| 7/7 | 4.15 pm | 6 | Network Data Analysis and Visualization | |
| 7/10 | 1 pm | 7 | Computational Complexity | HW3 |
| 7/10 | 4.15 pm | 8 | Data Structures Continued | |
| 7/12 | 1 pm | 9 | Gradient Based Algorithms | HW4 |
| 7/12 | 4.15 pm | 10 | Computational Methods and Algorithms Continued | |
| 7/14 | 1 pm | 11 | Computational Methods and Algorithms Continued | HW5 |
| 7/14 | 4.15 pm | 12 | Supervised versus Unsupervised Learning | |
| 7/17 | 1 pm | 13 | Introduction to Large Scale Data Sets | HW6 |
| 7/17 | 4.15 pm | 14 | Introduction to Parallel Computing | |
| **7/21** | **1 pm** | | **Final Project Due No Later Than 1pm** | |

## III. Course Materials

- Slides from class will be posted on **Blackboard**

## IV. Course Requirements

### 1. Homework (56%)

Each student will be required to complete seven homework assignments. In most cases, HW will involve programming exercises based on the material covered in lectures. Assignments should be submitted through Blackboard (as *iPython* notebooks).

Grading criteria for programming assignments:

- 5 – Code submitted, but there are errors running/executing the code.
- 6 – Code runs, but results are incomplete.
- 7 – Code runs, results are complete but some results are incorrect.
- 8 – Complete results, but there is room for improvement. (i.e., no comments, graphs are not labelled, no description of steps)
- 9 – Good analysis, all results well presented.
- 10 – Excellent, with some extensions considered or interesting issues identified (i.e., doing more than what has been asked).

Grading criteria is applied to each question separately. The grade for the programming assignment is the average grade across questions in that assignment.

### 2. Class Attendance and Participation (10%)

This will be determined based on your attendance and your overall contribution to the class. You are expected to come to class on time. Your instructor may cold call you to answer questions in class, and you are expected to be prepared to answer these. If you are not attentive in class (for example distracted by electronic devices) then you will not be able to answer questions when called upon.

Attendance will be taken in several class sessions. Each student will be expected to contribute to class discussion. The discussion may include questions about the material, the important points in the lectures/assignments, or suggestions for alternative approaches.

### 3. Final Project (30%)

A more extensive final project, along with written report, will be due on the last day of class. Students will be expected to agree with the instructor on the topic of the project by about halfway through the module. Please submit a copy of your code, your final report, and any relevant data by July 21st on Blackboard.

**Summary**

| | |
|---|---|
| Final Project | 33% |
| 7 HWs | 56% |
| Class Participation | 11% |

## V. Academic Honesty

Purdue University and Krannert School of Management guidelines on Academic Honesty will be strictly enforced: http://www.purdue.edu/odos/academic-integrity

## VI. Class Details

1. **Introduction**
    - Course Overview
    - Data Analytics Problems
    - Why Need Computational Tools?
    - Why Python?
        - Python Setup

2. **Data Structures**
    - Lists
    - Tuples and Sequences
    - Dictionaries and Hash Tables
    - Data Frames
        - Pandas Library

3. **Computing Summary Statistics**
    - Calculating Mean, Median, Std. Dev, etc.
    - Random Sampling
        - Numpy Library
        - Scipy Library

4. **Data Visualization I**
    - Drawing Informative (and attractive) graphs
    - Bar Charts and Histograms
    - Scatter and Bubble Plots
    - Heatmaps
        - Matplotlib Library
        - Seaborn Library

5. **Data Visualization II**
    - Dynamic and interactive data visualization in web browsers
        - Bokeh Library

6. **Introduction to Network Data Analysis and Visualization**
    - Centrality Measures
        - NetworkX Library

7. **Computational Complexity**
    - Order Notations
    - Linear, quadratic, logarithmic computational complexities

8. **Recursive Algorithms**
    - Recursive Procedures
    - Gradient Based Algorithms

9. **Computational Methods and Algorithms Continued**
    - Searching
    - Binary Search Tree

10. **Computational Methods and Algorithms Continued**
    - Sorting

11. **Data Structures Continued**
    - Sparse Matrix Storage

12. **Supervised versus Unsupervised Learning**
    - Estimating regression coefficients
    - Online linear regression algorithm

13. **Introduction to Large Scale Data Sets**
    - Hadoop
    - MapReduce

14. **Introduction to Parallel Computing**
    - mutliprocessing