PSY 394U: Data Analytics with Python

(Unique# 42755)

Spring 2019

Class Meets: 2-3:30pm, Tuesdays & Thursdays, SEA 2.116

Instructor: Satoru Hayasaka
Office: SEA 2.214
Email: hayasaka@utexas.edu
Phone: 512-475-6177

Office hours:
10-11:30am, Tuesdays & Thursdays
and by appointment

Course Description

***University Catalog Course Description*:** Examines analytical tools available in Python for various types of data. Subjects include multivariate statistics, machine learning, network data, and text mining. Previous Python experience strongly preferred.

What will I learn?

Upon completion of this course, students will be able to:

- Understand various data analytics methods, including machine learning, network analysis, and natural language processing. In particular, when and how such methods are used.
- Manipulate a data set using Python so that it can be analyzed by the analysis methods described in this course.
- Implement simple data analyses using the methods described in this course.
- Interpret their data analysis results appropriately.

Learning Outcomes:

1. Vocabulary and understanding of data analytics methods
2. Python coding skills for data analytics
3. Problem solving skills necessary to address a research question of interest through data analytics

This course DOES NOT:

- Provide in-depth details on a particular analytics technique.
- Explain all possible parameters, options, or methods associated with a particular analytic function / library.

How will I learn?

This course is presented as a hybrid of lectures and programming exercises. During each class, the instructor presents concepts for data analytics and associated resources in Python. The instructor also gives a number of small programming exercises throughout the class. Students will work on those exercises, and in some exercises, share their codes with the class.

There will be two mini-hackathons, or group programming projects, during the semester.

Pre-requisites for the course:

Some programming experience in Python. The instructor assumes that students are familiar with array data in NumPy and data frames in Pandas.

Most of the data analytics methods presented in this class are based on mathematical / statistical principles. Thus, students should have understanding of basic statistical concepts, such as mean, median, regression, and distributions.

Course Resources

Required Resources

Hardware:

A computer capable of running various analytics methods is required to complete in-class exercises, mini-hackathons, and homework assignments. Students are encouraged to bring a laptop to the class. Those who do not have access to a laptop, may use computers in the classroom. Please note that computers in the classroom are not always available.

Software:

If you have Python on your computer ...

Python: We use Python 3 (3.6 or later) in this class. We DO NOT use Python 2 (any version), as some syntax is different between Python 2 and 3.

Python libraries: You need to install the following libraries. Note that these libraries are for Python 3. Even if you have these libraries for Python 2, you need to install these for Python 3.

NumPy	SciPy	Matplotlib	Pandas
scikit-learn	networkX	NLTK	wordcloud

If you <u>DO NOT</u> have Python on your computer ...

Docker: If you do not have Python on your computer, then I ask you to install Docker on your computer. Docker is similar to a virtual machine (a computer inside your computer), but is much simpler and requiring far less resources to run than a virtual machine. Docker enables you to install and run a bundle of software packages and libraries known as a Docker image.

Docker image: A Docker image called sathayas/python-analytics-bundle is available from the Docker Hub web site. This Docker image includes Python 3, all the Python libraries listed above, as well as Jupyter notebook. In addition, Graphviz is also included in this Docker image.

Installation instructions on Docker and Docker image will be coverd during the first class.

Optional Resources

Textbooks: There is no required textbook for this class. However, the class is loosely organized based on the materials from these resources:

Machine learning:

Scikit-learn user's guide (http://scikit-learn.org/stable/user_guide.html)

Natural language processing:

Bird, Klein & Loper, *Natural Language Processing with Python*. Available online (http://www.nltk.org/book/).

Git: Git enables you to clone (or copy) codes, notes, and some data associated with this class from GitHub. Git is already pre-installed on Mac. For Windows users, you can download Git for free from its website (https://git-scm.com/downloads).

Jupyter Notebook: Jupyter Notebook enables you to read course notes and run Python codes on your web browser. Installation instructions and a brief demonstration of Jupyter Notebook will be presented during the first class. (Included in the Docker image)

Graphviz: Graphviz is a software package for visualizing networks and trees. It will be used to display decision trees. Graphviz is freely available (https://www.graphviz.org/download/). (Included in the Docker image)

Python Refresher: If you need a refresher course on Python (including NumPy and Pandas), you can read my course notes from my Intro Python class

https://github.com/sathayas/JupyterPythonFall2018

It should be noted that there will be no formal remedial tutorial on Python coding during the class.

Course Materials

Course Notes: Course notes are in Jupyter Notebook format, and available from the instructor's GitHub repository at:

https://github.com/sathayas/JupyterAnalyticsSpring2019

Students may clone the repository or download individual documents to be opened in Jupyter notebook on their computer. Alternatively, students can view the notes on their web browser. When viewed in a web browser, codes cannot be executed. Links to individual notes are provided on Canvas. *Please note that notes will be added and updated throughout the semester.*

Codes & Data: Some example codes and data used during the class, as well as sample solutions for in-class exercises and homework assignments, are available on GitHub at

https://github.com/sathayas/AnalyticsClassSpring2019

Students may clone the repository or download individual code or data file to be saved on their computer. Codes can also be viewed on a web browser. Codes are organized into folders for different modules. Links to folders are provided on Canvas.

Grading

Homework Assignments (60 points)

There will be five homework assignments throughout the semester. Homework assignments will be posted on Canvas. Each assignment involves writing of codes to analyze data. Students are expected to turn in the source code(s) associated with each assignment on Canvas.

- **Homework Assignment 1** (Mandatory) (15 points)
  Refresher. This assignment is to refresh your memory on NumPy, Pandas, and Matplotlib. This assignment is to ensure that you have sufficient Python coding background to follow the materials for the rest of the semester.

- **Homework Assignments 2-5** (15 points each)
  These assignments cover different topics covered during the class.

The grade for the homework assignments will be assessed by the following formula:

HW Score = Score (HW1) + Three highest scores (HW 2-5)

With the maximum possible score of 60 points.

Mini-Hackathons (30 points)

There will be two mini-hackathons during the semester, each accounting for 15 points. Each mini-hackathon is a group programming project by a team of 2-3 students. Two class periods will be devoted for each mini-hackathon. Students will be given a list of problems to be addressed prior to a mini-hackathon, and each student will choose a topic of interest. Those selecting the same topic will be organized into a team. The size of each team will be 2-3 students; if there is any uneven distribution of students, the instructor will merge / split teams into appropriate sizes. Each team will code collectively to address the problem, present their code and the solution to the problem, and submit the code.

Code Presentations (10+ points)

Students present their code during class exercises. Students can post their code using the Discussion function on Canvas during a class, by responding to a post by the instructor. Among posted submissions, one student is selected to present his/her code to the class. Students can earn:

6 points - First code presentation
3 points - Second code presentation
1 points - Third and subsequent code presentations

Note that you will not earn points just by posting your code on Canvas; you have to be selected to present your code to earn points. The instructor will ensure that different students have opportunities to present when selecting a submission. The priority will be given to students earning the most points for the code presentation. Depending on the number of presentations, a student may earn more than 10 points for code presentations. The excess points will be added during the calculation of the final grade. If a student's percentage points for the

final grade exceed 100 points, then the grade is capped at 100 points.

Final Grade

The final grade is based on the percentage of possible points from the homework assignments, mini-hackathons, and code presentations (capped at 100 points):

Points	Grade	Points	Grade
93-100	A	67-69.99	D+
90-92.99	A-	63-66.99	D
87-89.99	B+	60-62.99	D-
83-86.99	B	Below 60	F
80-82.99	B-		
77-79.99	C+		
73-76.99	C		
70-72.99	C-		

Policies

Auditing

Those who are interested in auditing this class may do so by requesting a permission from the instructor BEFORE attending the class. The instructor may grant access to Canvas for those auditing the class. The instructor reserves the right to revoke the permission to audit when a person auditing the class is deemed disruptive to the class (e.g., asking many questions not directly related to the class, engaging in activities unrelated to the class).

Attendance

Attendance is mandatory for the mid-term and final mini-hackathons. If you are unable to attend these due to religious holidays or extenuating circumstances (e.g., health issues, family emergencies), please consult the instructor as soon as possible so that an alternate arrangement can be made. The instructor will not track attendance for regular (non-hackathon) sessions. However, if you are not in the class, you will not be able to earn points for code presentations.

Late Assignments

Unless a student receives a prior permission from the instructor for a late submission, points will be deducted for assignments turned in late according to the following rule:

- Before the solution is presented and posted on GitHub – 1 point
- After the solution is presented in the class:

1 day late	2 points
2 days late	4 points
3 days late	6 points
More than 3 days late	8 points
More than a week late	15 points

If there is an extenuating circumstance (e.g., health problems, family emergencies, etc.) to prevent you from submitting a homework assignment, *to avoid reduced points*, please ask the instructor's permission for a late submission ***BEFORE*** the deadline. When there is a late submission with a prior approval, the instructor may delay posting the solution on GitHub.

Group Submissions

Students may work together on a homework assignment. If students do work together, the students in the same group may submit the identical codes. Each student is still responsible for submitting homework assignments individually on Canvas. When your homework assignment submission is part of a group submission, you must comment in your submission (either in the code itself or using the comment function in Canvas) whom you worked with. If nearly identical codes are submitted without noting any collaboration, such submissions are considered cheating. When students are working together, their group size may not exceed 4.

Academic Integrity

Plagiarism, cheating, collusion, and any other form of academic misconduct are considered a serious offence. The instructor will follow the definition of academic misconduct defined by Chapter 11, Student Discipline and Conduct, of the Institutional Rules on Student Services and Activities when determining whether such an infraction has taken place. Students can find materials on student conduct and academic integrity at

http://deanofstudents.utexas.edu/sjs/acint_student.php

Please note that the instructor will report any incident of academic misconduct. The sanction for an academic misconduct ranges from reduced points on assignments to dismissal from the university, depending on a student's prior records on academic misconduct.

Students with Disabilities and Different Learning Styles

This class respects and welcomes students of all backgrounds, identities, and abilities. If there are circumstances that make our learning environment and activities difficult, if you have medical information that you need to share with me, or if you need specific arrangements in case the building needs to be evacuated, please let me know. I am committed to creating an effective learning environment for all students, but I can only do so if you discuss your needs with me as early as possible. I promise to maintain the confidentiality of these discussions. If appropriate, also contact Services for Students with Disabilities, 512-471-6259 (voice) or 1-866-329- 3986 (video phone). http://ddce.utexas.edu/disability/about/

Course Schedule

***Note**: The course schedule is subject to change, depending on the amount of materials we can cover. It is your responsibility to note these changes when announced (although I will do my best to ensure that you receive the changes with as much advanced notice as possible).*

Date		Topic	Homework
1/22	Unsupervised learning	Overview of the course, Docker, Jupyter	
1/24		Principal component analysis (PCA)	
1/29		Principal component analysis (PCA) (Cont'd)	HW1 Due
1/31		Factor analysis, independent component analysis (ICA)	
2/5		Clustering	
2/7		Clustering (Cont'd)	
2/12	Supervised learning	Linear discriminant analysis	
2/14		Logistic regression	
2/19		Support vector machine (SVM)	HW2 Due
2/21		Support vector machine (SVM) (Cont'd)	
2/26		Nearest neighbors	
2/28		Decision trees	
3/5		Neural networks	
3/7		Cross validation, model evaluation	
3/12		Mid-term mini-hackathon (day 1)	
3/14		Mid-term mini-hackathon (day 2)	
3/19		***Spring break – No class***	
3/21		***Spring break – No class***	
3/26	Network analysis	Network data basics	HW3 Due
3/28		Network statistics	
4/2		Centrality	
4/4		Centrality (Cont'd)	
4/9		Network community	
4/11		Network community (Cont'd)	
4/16	Natural language processing	Text processing	
4/18		Text processing (Cont'd)	
4/23		Corpora, WordNet	HW4 Due
4/25		Word cloud	
4/30		Text classification	
5/2		Text classification (Cont'd)	
5/7		Final mini-hackathon (day 1)	
5/9		Final mini-hackathon (day 2)	
5/14			HW5 Due