MIS284N - Big Data and Distributed Programming

> *I am pleased to welcome you to your first course in Computer Engineering. I charge you to, ask questions, be curious, have fun learning and, conduct yourself with honor. I will strive to give you my best!*

Course Overview

This course covers a range of topics required for developing modern applications that operate over vast data sets that are potentially distributed in nature. The course covers a range of alternative technologies and architectures for working with big data, examining the pros and cons of the different approaches. The course also covers some unifying distributed programming fundamentals that are pervasive across the different technologies. New and emerging trends in Big Data, like Streaming-data anaysis, Data Lakes and Data Ops. Students will also get hands on experience through a series of small programming assignments exercising modern concepts and tools, Specifically, the assignments will use open source platforms like Apache Hadoop (MapReduce) and Apache Spark and Storm and a Cloud storage platform (e.g., Amazon AWS, Microsoft Azure, Oracle Bare Metal Cloud, or Google Cloud Platform). The programming projects will be in Python, which is fast emerging as preferred language for data collection, mining and visualization of big data.

Problem Sets

The Problem Sets are Jupyter notebooks with a set of short problems that you are asked to solve. They are intended to help you better understand the concepts covered in class. There will be four of these.

Programming Projects

There are three significant programming projects which will be in teams of two. They explore the specific concept by solving a larger problem. The three topics for the projects are i. Solving a large dataset problem using Map-Reduce on Apache Spark, ii. Solving a large dataset problem using the Machine Learning process, iii. Solving a large dataset problem using AWS (EC2).

Late Policy

Problem Sets must be turned in on the due midnight on due date (usually 10 days). There are no late exceptions for Problem Sets. Programming assignments are due midnight on the due date. You are allowed to submit programming assignments 1 and 2 (not 3) late with a 10% deduction per day up to a maximum of 2 days.

Additional Details

The University of Texas at Austin provides, upon request, appropriate academic adjustments for qualified students with disabilities. For more information, contact the Office of the Dean of Students at 471-6259, 471-4241 TDD, or the College of Engineering Director of Students with Disabilities, 471-4321.

Quick Facts. . .

Classroom: EER 1.516

Class Time: TTh 9-11am

Pre-requisites:
Programming (in Python) experience is expected

Office Hours:
MW 2:00-4:00pm (EER 5.824)
TA Office Hours on Canvas

Grading Criteria:

Assessment	Percentage
Problem Sets (4)	40%
Projects(3)	60% (20% each)

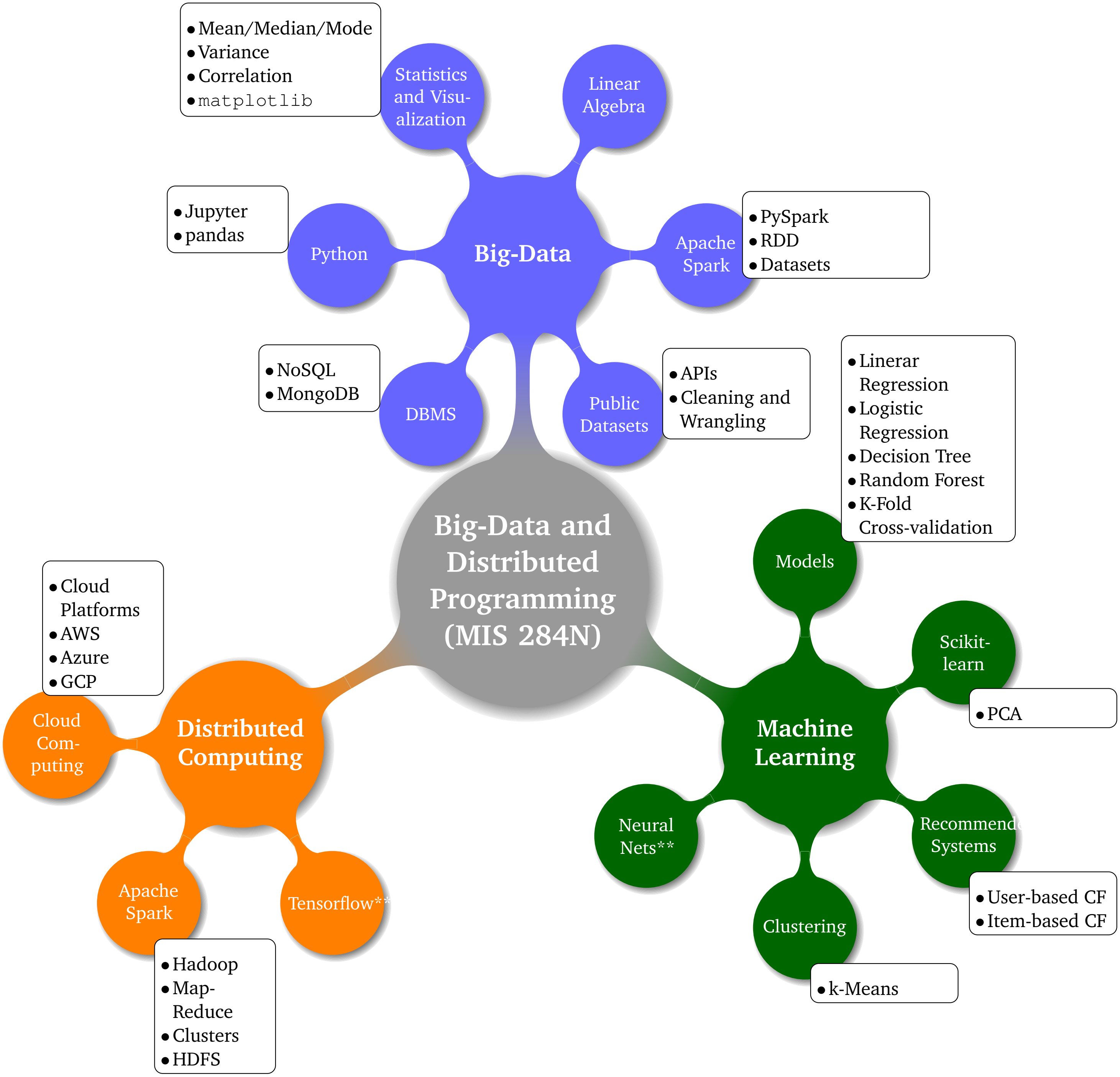
Teaching Assistants:
Aastha T.: tripathiaastha68@gmail.com

Important Due Dates:

Problem Set 1	Monday 9/10
Project 1	Monday 9/17
Problem Set 2	Monday 9/24
Project 2	Monday 10/1
Problem Set 3	Monday 10/8
Problem Set 4	Monday 10/15
Project 3	Monday 10/19

Class Website:
`http://canvas.utexas.edu/`

Concept Map



Big-Data and Distributed Programming (MIS 284N)

Big-Data

- Statistics and Visualization
  - Mean/Median/Mode
  - Variance
  - Correlation
  - `matplotlib`
- Linear Algebra
- Python
  - Jupyter
  - pandas
- Apache Spark
  - PySpark
  - RDD
  - Datasets
- DBMS
  - NoSQL
  - MongoDB
- Public Datasets
  - APIs
  - Cleaning and Wrangling

Machine Learning

- Models
  - Linerar Regression
  - Logistic Regression
  - Decision Tree
  - Random Forest
  - K-Fold Cross-validation
- Scikit-learn
  - PCA
- Recommend Systems
  - User-based CF
  - Item-based CF
- Clustering
  - k-Means
- Neural Nets**

Distributed Computing

- Cloud Computing
  - Cloud Platforms
  - AWS
  - Azure
  - GCP
- Apache Spark
  - Hadoop
  - Map-Reduce
  - Clusters
  - HDFS
- Tensorflow**

**: Advanced topic will be covered if time permits

Tentative Schedule:

Tuesday	Thursday
Aug 28th (No Class)	30th **1** What is Big Data?
Sep 4th **2** Python for Big Data	6th **3** Statistics and Visualization
11th **4** Apache Spark	13th **5** Apache Spark
18th **6** Machine Learning	20th **7** Models
25th **8** Performance Metrics	27th **9** Libraries (scikit-learn)
Oct 2nd **10** Clustering and Recommender Systems	4th **11** Distributed Computing
9th **12** Cloud Computing	11th **13** Cloud Platforms - AWS,Azure, GCP
16th **14** AWS	18th **15** Buffer

Academic Honesty

Integrity is a crucial part of your character and is essential for a successful career. We expect you to demonstrate integrity in this course and elsewhere. In particular, your assignments must represent your own work and understanding. Academic misconduct such as plagiarism is grounds for failing the class.The following guidelines apply unless an assignment specifically states otherwise. If you have any questions about acceptable behavior, please ask the course staff. We are happy to answer your questions! You are encouraged to talk to your classmates about solution ideas, and you may reuse those ideas, but you may not examine nor reuse any other student's code. You are not allowed to copy code from any source — other students, acquaintances, the Web, etc. (Copying is forbidden via cut-and-paste, via dictation or transcription, via viewing and memorizing, etc.) You are encouraged to use books, the Internet, your friends, etc. to get solution ideas, but you may not copy/transcribe/transliterate code: get the idea, close the other resource, and then (after enough time that the idea is in your long-term, not short-term, memory) generate the code based on your own understanding.

Examining other people's code

You may sometimes find it useful to do a web search to find snippets of code that perform some particular operation, and you may subsequently paste this code into your own program. This can be an acceptable short-term strategy if it helps you get past a particular roadblock. However, you must later go back, remove the code you did not write yourself, and write the replacement on your own, from scratch. It is your responsibility to understand everything that you turn in. We reserve the right to ask you to explain any part of your homework assignment. If you are not able to explain what it means and why you chose it, that is presumed evidence of copying/cheating.

Later, when you are writing your own programs after you complete this course and your degree, it's fine to copy others' code if the license associated with the code permits such use. However, in your future career, please remember two things:

1. It is your ethical duty to properly cite the source of any code that you did not write yourself. Give credit where credit is due.
2. You should still understand any code that you copy. Otherwise, if and when the code does not work (for example, if the original author made an assumption that is not true in your program), you will lose more time debugging than you saved by copying.

The key idea is that we want you to understand. Sometimes you can achieve that by examining and understanding other people's code. But you can never achieve that by copying alone.

In summary, we are committed to preserving the reputation of your UT degree. To guarantee that every degree means what it says it means, we must enforce a strict policy on academic honesty: every piece of work that you turn in with your name on it must be yours. As an honest student, you are responsible for enforcing this policy in three ways:

1. You must not turn in work that is not yours, except as expressly permitted by the instructors. Specifically, you are not allowed to copy someone else's program code. This is plagiarism.
2. You must not enable someone else to turn in work that is not his or hers. Do not share your work with anyone else. Make sure that you adequately protect your files. Even after you have finished a class, do not share your work or published answers with students who come after you. They need to do their work on their own.
3. You must not allow someone to openly violate this policy because it diminishes your effort as well as that of your honest classmates.

Students who violate University rules on scholastic dishonesty in assignments or exams are subject to disciplinary penalties, including the possibility of a lowered or 0 grade on an assignment or exam, failure in the course, and/or dismissal from the University. Changing your exam answers after they have been graded, copying answers during exams, or plagiarizing the work of others will be considered academic dishonesty and will not be tolerated. Plagiarism detection software will be used on the programs submitted in this class.

If cheating is discovered, a report will be made to the Dean of Students. Allegations of Scholastic Dishonesty will be dealt with according to the procedures outlined in Appendix C, Chapter 11, of the General Information Bulletin, `http://www.utexas.edu/student/registrar/catalogs/`