# Data Science

- **Course:** CSE 405/605 - Data Science
- **Schedule:** Tuesday and Thursday 12:30 pm - 1:45 pm
- **Instructor:** Dr. Somya D. Mohanty
- **Location:** 303 Petty

- **Class Discussions:** https://discord.gg/ktdrsUH4Fd (use #csc-405-605 channel for class discussions)
- **Office Hrs:** Thursday 2:00 pm - 3:00 pm via Zoom only (email for appointment and Zoom link)
- **Email:** [sdmohant@uncg.edu](mailto:sdmohant@uncg.edu and mohanty.somya@uncg.edu)

- **Online Attendance:** Zoom link - https://uncg.zoom.us/j/3634402596?pwd=N01WaW9QM0c2VlpqUjlPZk0xRjQ5Zz09. Note, you have to let me know the reason for missing in person class, else you will **not** be admitted into the session. Only medical excuses and covid related cases allowed.

## Course Description

In a world with ever increasing data generated both by humans and machines alike, the field of computer science has seen a transition from computation-intensive solutions to data-intensive ones. Often in such a scenario, solutions to real-world problems can be derived/learned by analyzing disparate, complex and messy datasets using Data Science methods and approaches.

This course is highly interactive, and will explore the theories, techniques, and the tools necessary to gain insights from such datasets. Using a problem-based learning philosophy, students are expected to make use of such technologies to design data solutions that can process and analyze real-world datasets for a variety of scientific, social, and environmental challenges.

The core topics addressed by the course will be:

- Programming with Data
- Data Mining, Munging, Wrangling
- Statistics, Analytics, Representation, Visualization
- Introduction to Applied Machine-Learning

## Prerequisites

CSC 339 (Programming Languages) OR Programming experience (Instructor Permission Required)

## Textbooks

There is no required text for the course. Class slides will be available for download. Suggested textbooks are: 1) Building Machine Learning Systems with Python (Richert and Coelho), 2) Data Science from Scratch (Joel Grus)

# Course Overview

This course is highly interactive and based on the problem-based learning philosophy; students are expected to make use of said technologies to design highly scalable systems that can process and analyze real-world datasets for a variety of scientific, social, and environmental challenges.

# Course Topics and Schedule (Tentative)

1. **Introduction to Data Science: (Week 1)**
    - Class Syllabus and Introduction
    - Class Project discussion and assignment
2. **Startup Tools and Programming (Weeks 2-3)**
    - Programming
        1. Re/Introduction to Python
        2. IPython, IPython-Notebook
    - Data Science Reproducibility
        1. Setting up your Repository – Data, Code, and Documentation
        2. Using Version Control with Git
    - Final Project Discussions - Goals and Requirements
3. **Data Munging, Wrangling, Cleaning (Week 4-5)**
    - Data Structures for Data Science
    - Data Manipulation
        1. Selection - Indexing
        2. Handling Missing Data
        3. Aggregation
        4. Descriptive Statistics
        5. Merging / Join
        6. Working with Date-Time
    - ***Project Review - Stage I***
4. **Data and Statistics (Week 6-9)**
    - Distributions
    - Point Estimates
    - Statistical Hypothesis Testing
    - Correlation
    - Distribution Estimators
        1. MoM, MLE, KDE
    - ***Project Review - Stage II***
5. **Introduction to Applied Data Modeling: (Weeks 10-12)**
    - Applied Machine Learning
    - Regression and Feature Selection
    - Bias versus Variance

    - Clustering and Dimensionality Reduction
    - Validation and Model Performance
    - ***Project Review - Stage III***
6. **Data Visualization (Week 13-14)**
    - Graph Generation
        1. Types of Graphs
        2. Customizing Plots
        3. Visualizing Errors
        4. Interactive / Dynamic Graphs
    - Visualization Best Practices
    - ***Project Review - Stage IV***

- **Project Presentations: (Week 15 – Final's Week)**

Class slides and ipython notebooks will be available here.

# Grading

| **Grade** | **Max% to Min%** |
|---|---|
| **A** | 100% to 94% |
| **A-** | < 94% to 90% |
| **B+** | < 90% to 87% |
| **B** | < 87% to 84% |
| **B-** | < 84% to 80% |
| **C+** | < 80% to 77% |
| **C** | < 77% to 74% |
| **C-** | < 74% to 70% |
| **D+** | < 70% to 67% |
| **D** | < 67% to 64% |
| **D-** | < 64% to 60% |
| **F** | < 60% to 59% |

1. **Class / Homework Assignments (4):** 30%
    - https://github.com/UNCG-CSE/CSC-405-605_Spring_2022/tree/master/Assignment

    Four programming based in-class homework assignments will be given covering the utilization of the tools learned in class. Absolutely no collaboration on assignments. Students have to upload (Notebooks) individual assignments to GitHub. Listed below are the homework assignments for the class:

    2. Create a IPython notebook of random team generator based on a student list.

3. Evaluate the condition “Women and Children First”, in survivors’ dataset of Titanic.
4. Create a IPython notebook for the best “Date-Night” movie based on Movies, Ratings, and User database.
5. Visualize the results of the Titanic dataset: 1) Show survival rate by gender, ticket-class, and age. 2) Perform a Kernel Density Estimation and Box Plot of the passenger Ticket cost.

2. **Final Project:** 70%
    - https://github.com/UNCG-CSE/CSC-405-605_Spring_2022/tree/master/Project

The final project of the class will focus on the end to end development of an analytical model. The project will be split into four stages:

- Stage I Data/Project Understanding,
- Stage II Data Modeling,
- Stage III Distributions and Hypothesis Testing,
- Stage IV Basic Machine Learning, and
- Stage V Visualization and Dashboard.

This will be a team-based effort, where in first week of the course the students split into teams of 4-5 students. After completing each stage, the teams will have to give a short presentation (3-5 mins) and a report (1 page) of their progress with the project. The projects will be open-source and the teams will have to use GitHub as their code repository. Upon completion of the project the teams will present their software along with the results in form of a presentation (20 minutes).

6. Each Stage of the Final Project has 100 points. They will be equally weighted for the project final score.
    1. Each stage has deliverables of:
        1. Report
        2. Code Jupyter/IPython Notebooks
        3. Presentation
    2. To get the full points in each stage you need to finish all of the deliverables.
7. ***Graduate Students Only:*** Stage IV has 80 points for your project and 20 points for project report. Minimum 5 pages for single author, 8 for 2 authors, and 12 for 3 authors (figures and references included). Template:. Example: (Due: 12/02/2021)

**Total: 100%**

**Deadlines**

| Category | Sub-Category | Deadline |
|---|---|---|
| **Assignment** | * Assignment 1 | 01/25/2022 |
| | * Assignment 2 | 02/22/2022 |

| Category | Sub-Category | Deadline |
|---|---|---|
| | * Assignment 3 | 03/15/2022 |
| | * Assignment 4 | 04/05/2022 |
| **Project** | * Stage I | 02/08/2022 |
| | * Stage II | 03/01/2022 |
| | * Stage III | 03/22/2022 |
| | * Stage IV | 04/12/2022 |
| | * Stage V | 04/26/2022 |

**Submissions**

- **Assignments**:
  - Create a ***private Github repository*** (under your own account) and call it --- CSC-405-605_Spring-2022_Assignment.
  - Send me and our TA access to the repository,
    - My username: somyamohanty
    - Our TA is: ChanduMalgari (Chandra Shekhar Reddy Malgari)
  - Create a folder within the repository /Assignment_1
    - Create two sub-folders /src and /data
    - Work on your assignment (under /src)
      - IPython notebook only(.ipynb). Python will not do (.py).
      - Comment your code appropriately in Markdown.
  - Enter the link to your assignment solution in the assignment text entry (on canvas) once you are done with your solution.
  - Your notebook should contain the output of your cells. If there is no output rendered we will not grade it.
  - No collaboration at all in assignments
- **Project**:
  - Your code and documentation will reside in a ***project repository***.
  - The structure of the repository should be maintained as such.
    - /src - code and notebooks
      - /team
        - /stage_X
      - /member
        - /{member_name}
          - /stage_X
    - /data - data folder for the repository
      - /stage_X
    - /utility - utility or scripts
    - /doc - documentation - project reports and presentations
    - Readme.MD - Description of project, deliverables, team members (see Stage I for details)
    - all src files (notebooks) should use relative path.

- Each project has separate deliverables - notebooks need to be updated into the repository for grading. We will grade the status of repository at the time of deadline.
- Each team makes a ***recorded presentation*** of their project stage and uploads it to canvas. Top presentations will be discussed in class.
- No collaboration on member tasks.

## Communication

Discord channel for class discussions and team creation: https://discord.gg/ktdrsUH4Fd. The channel should be used for discussing general questions related to assignments and projects. Use this channel to ask questions and find anwsers to already responded quesitons. If the question has been already answered in the channel, I will not be responding to emails. Emails are a one-to-one conversation which takes a lot of time hence the channel is there to broadcast information and have more community oriented discussion. Do not share code or screenshots of code in the channel. Email should be the last step and can be used to ask student specific questions.

## Presentation Pointers

- You are going to be reviewed on the following criterion:
  - Criterion 1 (**C1**): Organize/Create information/slides in a manner appropriate for the intended audience
  - Criterion 2 (**C2**): Deliver information in a manner appropriate for the intended audience
  - Criterion 3 (**C3**): Relate to the intended audience
- For each criterion the evaluations/scoring are based on (higher the better):
  - **4 - Exceeds Criteria:** Excellent organization; information is well organized. Clear introduction; main points well stated and argued, with smooth transition to next point. Clear summary and conclusion.
  - **3 - Meets Criteria:** Satisfactory organization; clear introduction; main points are well stated; some transitions are somewhat sudden. Clear conclusion.
  - **2 - Progressing to Criteria:** Information is somewhat organized. Audience may have difficulty following presentation in areas.
  - **1 - Below Expectations:** Presentation is unorganized. Introduction unclear. Audience has difficulty following presentation. Presentation contains abrupt jumps; some of the main points and conclusion are unclear.

## Project Teams:

**Team 1: https://github.com/UNCG-CSE/Spring-22_COVID-Team_1**

- Meghna Nayal, meghnauncg
- Vamshi Krishna Edamadaka, vk-uncg
- Vrinda Prabhakaram Ganti, gantivrinda
- Yamini Nayal, ynayal

**Team 2:** **https://github.com/UNCG-CSE/Spring-22_COVID-Team_2**

- Sage Bonfield - wayTooMuchSauce
- Logan Whitfield - lww1117
- Jacky Luo - j4ckyluo
- Alyiah Proctor - Alyiahp
- Anne Nguyen - annednguyen00

**Team 3:** **https://github.com/UNCG-CSE/Spring-22_COVID-Team_3**

- Aman Tej Vidapu, Amantejv
- Saipavan Tadikonda, saipavantadikonda
- Viveka Erram, vivekareddy
- Lahari Chilakuri, lchilakuri
- Varsha Veeramaneni, VarshaRaoV

**Team 4:** **https://github.com/UNCG-CSE/Spring-22_COVID-Team_4**

- Aditi Darandale, aditidarandale06
- Manish Shah, manishshah1698
- Priyanka Budhavi, priyankabudavi
- Kyle Killworth, krkillworth
- Reetika Sarkar, rsarkar2

# Academic Honesty Policy

The instructor will deal strictly with any violations of academic honesty and integrity in this course. See Academic Integrity Policy (Link). for more details. ***Absolutely no discussion, collaboration, copying, and sharing on assignments. This includes coping from the internet. Any student who violates this policy will receive "F" in the course. The instructor will report the case to the university.***

# Attendance Policy

Attendance is required for all the class meetings. If you will be absent for any class it is your responsibility to catch up on class materials.

# Special Needs and/or Disabilities

Students with disabilities should have documentation from the Office of Accessibility Resources & Services (Link). This documentation should be provided to the instructor for review. In the case of major provisions such as separate testing environment or test-readers, the student must make arrangements with Office of Accessibility Resources & Services so that suitable accommodations can be provided.

# COVID Statement

As we return for spring 2022, all students, faculty, and staff are required to uphold UNCG's culture of care by actively engaging in behaviors that limit the spread of COVID-19. These actions include, but are not limited to:

- Following face-covering guidelines
- Engaging in proper hand-washing hygiene
- Self-monitoring for symptoms of COVID-19
- Staying home when ill
- Complying with directions from health care providers or public health officials to quarantine or isolate if ill or exposed to someone who is ill
- Completing a self-report when experiencing COVID-19 symptoms, testing positive for COVID-19, or being identified as a close contact of someone who has tested positive
- Staying informed about the University's policies and announcements via the COVID-19 website

Instructors will have seating charts for their classes. These are important for facilitating contact tracing should there be a confirmed case of COVID-19. Students must sit in their assigned seats at every class meeting. Students may move their chairs in class to facilitate group work, as long as instructors keep seating chart record.Students should not eat or drink during class time.

A limited number of disposable masks will be available in classrooms for students who have forgotten theirs. Face coverings are also available for purchase in the UNCG Campus Bookstore. Students who do not follow masking requirements will be asked to put on a face covering or leave the classroom to retrieve one and only return when they follow the basic standards of safety and care for the UNCG community. Once students have a face covering, they are permitted to re-enter a class already in progress. Repeated issues may result in conduct action. The course policies regarding attendance and academics remain in effect for partial or full absence from class due to lack of adherence with face covering and other requirements.

For instances where the Office of Accessibility Resources and Services (OARS) has granted accommodations regarding wearing face coverings, students should contact their instructors to develop appropriate alternatives to class participation and/or activities as needed. Instructors or the student may also contact OARS (336.334.5440) who, in consultation with Student Health services, will review requests for accommodations.

# Super Useful Links :)

### Jupyter Notebooks

- How To Enhance Jupyter Notebooks for Data Science?
- 28 Jupyter Notebook Tips, Tricks, and Shortcuts
- Optimizing Jupyter Notebook: Tips, Tricks, and nbextensions
- 28 Jupyter Notebook Tips, Tricks, and Shortcuts

### Exploratory Data Analytics:

- A Gentle Introduction to Exploratory Data Analysis
- 7 Steps to Mastering Data Preparation with Python
- Speed Up Your Exploratory Data Analysis With Pandas-Profiling
- Exploratory Data Analysis (EDA) and Data Visualization with Python

### Feature Engineering:

- Machine Learning with Kaggle: Feature Engineering
- Data cleaning and feature engineering in Python
- Feature Engineering Data Science Handbook
- A Hands-On Guide to Automated Feature Engineering using Featuretools in Python
- Feature Engineering Cookbook for Machine Learning

### Missing Value Analysis and Cleaning:

- Cleaning and Prepping Data with Python for Data Science — Best Practices and Helpful Packages
- Working with Missing Data in Pandas
- Data Cleaning with Python and Pandas: Detecting Missing Values
- How to Handle Missing Data with Python

### Pandas and Big Data:

- Why and How to Use Pandas with Large Data
- Using Pandas with Large Data Sets in Python
- Optimizing the size of a pandas dataframe for low memory environment
- Making DataFrame Smaller and Faster
- Reducing DataFrame memory size by ~65%
- Dask: Scalable analytics in Python