Syllabus

Welcome to Data Management for Data Science! This course covers data processing and management tools like relational databases and NoSQL for processing large volumes of data. Data engineering techniques like data lakes and warehouses, ETL vs ELT, data mesh, and data pipelines will be covered. NLP concepts like vector databases and Retrieval-Augmented Generation (RAG) will be covered, along with predictive analysis topics like time series analysis, boosting techniques like XGBoost, LightGBM. Visualization techniques like t-SNE, data storytelling, and sunburst plots will be covered. Majority of the programming will be in Python.

Revisions to Syllabus

- 09/09/2024: Added directions for accessing O'Reilly text books

Course Instructor

- Dr. Meenakshi Syamkumar (Teaching Faculty - Department of Computer Sciences) ms@cs.wisc.edu

Lecture (Meeting Time and Location)

- LEC001 Sewell Social Sciences 5208 01:20 PM - 02:10 PM
- We meet 3 times a week -- see the lecture schedule here.
- I'll ask questions during lecture via TopHat. Answering questions (and getting them correct) will help your participation score, though perfect attendance is not necessary for full credit.

Instructional Modality

- LEC001: in-person

Learning Objectives

- Manage large datasets using SQL and NoSQL databases and write programs to analyse the datasets
- Understand data integration techniques like ETL, ELT
- Extract and optimize data into a data lake
- Understanding critical role of semantic layers in data management
- Demonstrate competencies with tools like MongoDB, Apache Iceberg, Snowflake
- Understand and apply predictive analysis techniques like timeseries analysis, and boosting
- Demonstrate visualization competencies for creating data stories, dashboards

Readings

We'll be learning about many different data management systems, and so no textbook closely corresponds to the lecture content. Thus, attending lectures and taking notes will be your primary resource.

We will have recommended (though optional) readings.

Here are some of the main texts we'll reference this semester:

- Fundamentals of Data Engineering by Joe Reis & Matt Housley
- Big Book of Data Engineering by databricks
- Data Engineering with Python by Paul Crickard
- Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems (1st edition), by Martin Kleppmann

You can read O'Reilly text books free online via the Madison Public Library. You just need to do the following:

1. get a library card (free)
2. sign into the O'Reilly collection with your card number
3. search for the assigned book

Communication

We message the class regularly via Canvas announcements. We recommend updating your Canvas settings so that the "Announcement" option is "Notify immediately" so that you don't miss something important.

See the help page for details about how to contact us.

We have various forms for us to leave (optionally anonymous) feedback, report lab attendance, and thank TAs.

Course Components

Grading breakdown

- Midterm (15%)
- Final (20%)
- 12 quizzes (12% total)
- 8 programming projects (6% each, 48% total)
- participation (5% total)

At the end of the semester, you'll have a score out of 100, which will be mapped to a letter grade (see below).

Letter Grades

At the end of the semester, we will assign final grades based on these thresholds:

- 93% - 100%: **A**
- 88% - 92%: **AB**
- 80% - 87%: **B**
- 75% - 79%: **BC**
- 70% - 74%: **C**
- 60% - 69% **D**

I will reserve my decision on rounding decimal points to the end of the semester.

Exams

These will be multiple choice exams taken in person. The midterm and the final will be at a different location (to be announced).

Midterm will be held on Wednesday, October, 16th

Quizzes

There will be a short Canvas quiz due at the end of most Wednesdays. Make sure you know the rules regarding what is allowed and what is not.

Allowed

- however much time you need
- discussing answers with classmates who are taking the quiz **at the same time**
- referencing texts, notes, or provided course materials
- searching online for general information
- running code

NOT allowed

- taking it more than once
- discussing answers with anybody outside of the course
- discussing with classmates who have already completed the quiz when you haven't completed it yourself yet
- posting anything online about the quizzes
- using such material potentially posted by other students who broke the preceding rule
- getting TA/instructor help on quiz questions prior to the quiz deadline

Projects

See project policies here.

Participation

Some of the things that count towards participation:

- TopHat (most important)
- filling class surveys
- accepted pull requests fixing issues with project specifications
- instructor endorsed piazza contributions (be sure to use your NETID@wisc.edu email so we can identify you)
- other...

Academic Misconduct

Code copying between students is not allowed in this course, except between project partners. Copying includes emailing, taking photos, looking while typing line by line, etc. Copying code then changing it is still copying and thus not allowed. Lock your compute when it's not attended.

Be sure to read and understand the full project collaboration policies here.

Citing ChatGPT (or other LLMs): it's allowed with proper citation (see above link for details).

Citing Online Resources: you can copy small snippets of code from stackoverflow (and other online references) if you cite them. For example, suppose I need to write some code that gets the median number from a list of numbers. I might search for "how to get the median of a list in python" and find a solution at https://stackoverflow.com/questions/24101524/finding-median-of-list-in-python.

I could (legitimately) post code from that page in my code, as long as it has a comment as follows:

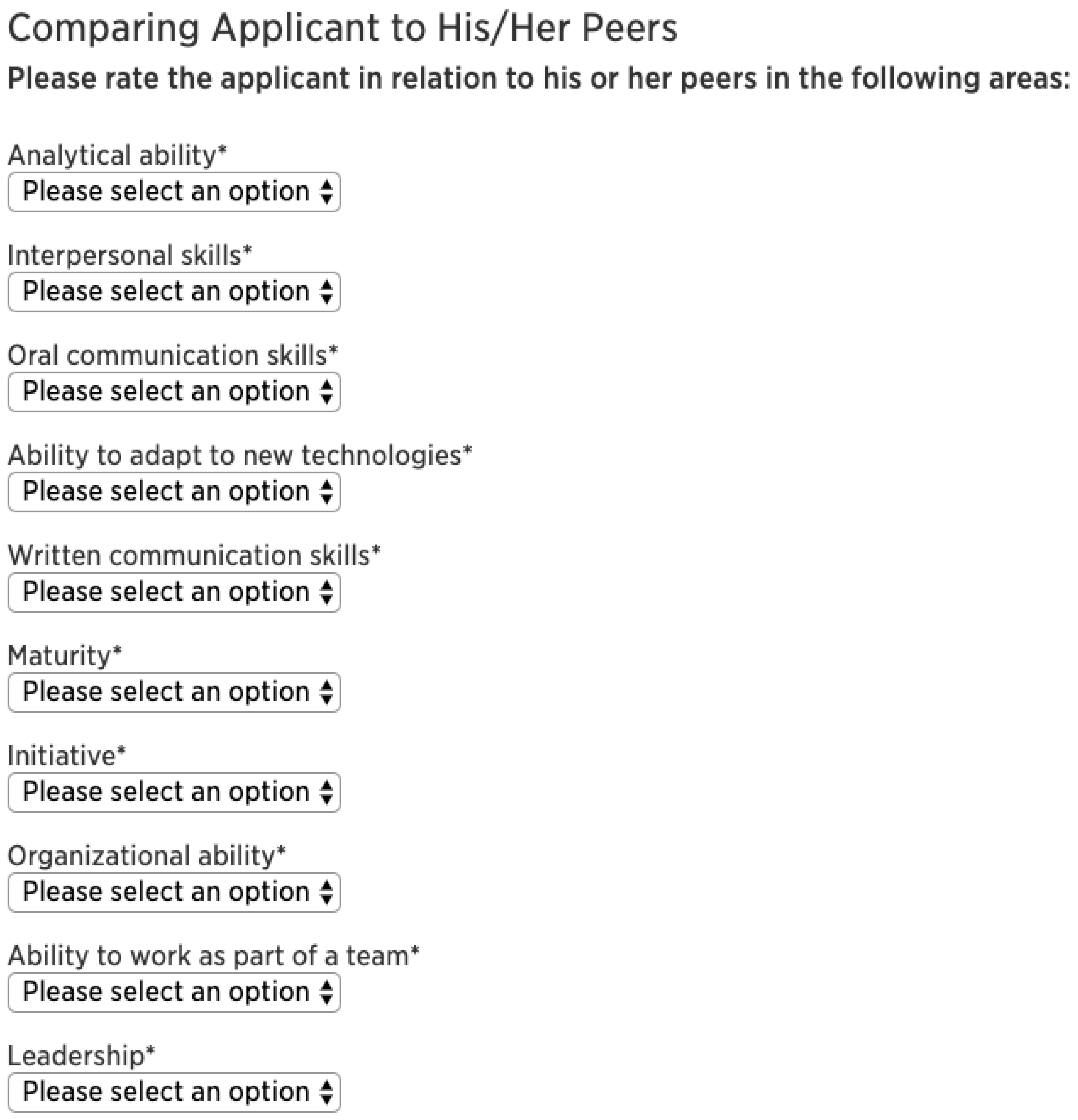
```
# copied/adapted from https://stackoverflow.com/questions/24101524/finding-median-of-list-in-python
def median(lst):
  sortedLst = sorted(lst)
  lstLen = len(lst)
  index = (lstLen - 1) // 2

  if (lstLen % 2):
    return sortedLst[index]
  else:
    return (sortedLst[index] + sortedLst[index + 1])/2.0
```

In contrast, copying from a nearly complete project you find online (that accomplishes what you're trying to do for your project) is not OK. When in doubt, ask us! The best way to stay out of trouble is to be completely transparent about what you're doing.

Recommendation Letters

Earning a recommendation letter is much harder than earning an A in this course. At a minimum, I'll want to see you doing something complex and interesting beyond the assingments. For a typical letter, I'll have collaborated with a student on some project for multiple months, with many iterations of feedback.

Most grad schools require recommenders to fill long forms rating students on various abilities (see an example below). Make sure that if you're asking me, I would be able to fill such a form without needing to put "I don't know" as my answer to many of the questions.



Comparing Applicant to His/Her Peers

Please rate the applicant in relation to his or her peers in the following areas:

Analytical ability*
Please select an option ⬍

Interpersonal skills*
Please select an option ⬍

Oral communication skills*
Please select an option ⬍

Ability to adapt to new technologies*
Please select an option ⬍

Written communication skills*
Please select an option ⬍

Maturity*
Please select an option ⬍

Initiative*
Please select an option ⬍

Organizational ability*
Please select an option ⬍

Ability to work as part of a team*
Please select an option ⬍

Leadership*
Please select an option ⬍