

SIADS 542 Course Syllabus

How to Get Help

If you have questions concerning the degree program, encounter a technical issue with Coursera, or issues using Slack, please submit a report to the ticketing system at umsimadshelp@umich.edu.

If you have an issue specific to the Coursera environment, you can also begin a [live chat session](#) with Coursera Technical Support (24/7) or view [Coursera troubleshooting guides](#). (you may be asked to log in to your Coursera account).

For questions regarding course content, refer to the Communications Expectations section below.

Course Overview

Students will learn how to correctly apply, interpret results, and iteratively refine and tune supervised machine learning models to solve a diverse set of problems on real-world datasets. Application is emphasized over theoretical content. The supervised learning course is an essential part of the core MADS machine learning series: its concepts, algorithms, and evaluation methods are used heavily throughout the following machine learning courses that include: unsupervised learning, deep learning, and machine learning pipelines.

Note: This course is based on material that also appears in the Supervised Machine Learning course in the Applied Data Science with Python Specialization on Coursera. A number of the weekly assessments are different in the MADS version, but some overlap may remain in a few cases. If you've already taken the Specialization version of this course, please contact the instructor if you would like the option of doing an alternate assignment for a given week. This course also contains significant content on additional topics not available in the Specialization.

Prerequisites

In order to be successful in this course, you must know how to program in Python and be familiar with the numpy and pandas Python libraries for data manipulation, and matplotlib for plotting.

Instructor and Course Assistance

Instructor: Jinseok Kim, jinseokk@umich.edu

Lecturer: Toby Kemp, tobyk@umich.edu

Lecturer: Saurabh Budholiya, srbh@umich.edu

Course Communication Expectations

Slack is the preferred communication tool for this course. If you have questions about course content (e.g. lecture videos or assignments), please make sure to use Slack. Instructor and course assistant response time to Slack messages will aim to be within 24 hours, Monday-Friday.

Please try to monitor the Slack channels for the course regularly.

Personal communication that may involve sensitive information may be emailed directly to the instructor or course assistant. If you email the instructor or course assistant, please include SIADS-542 in the email subject. Instructor and course assistant response time to email messages will try to be within 24 hours during the week.

Office Hours are held on:

- Jinseok Kim, Thursdays at 12:00 PM (noon) EST
- Toby Kemp, Thursdays at 6:00 PM EST
- Saurabh Budholiya, Fridays at 9:00 AM EST

Office hour sessions will be recorded for the benefit of students who are unable to join at these times.

Technology Requirements

The course programming will be based on Jupyter notebooks and Python 3.x.

Required Textbook

This course will use the following textbook as a reference and source of examples: Introduction to Machine Learning with Python, by Andreas C. Müller and Sarah Guido (O'Reilly Media)

This text is available under Resources in the Coursera shell (as a PDF file). It is also free online via the [University of Michigan Library](#):

1. On the Welcome! screen, choose "Select your institution" to open the menu and select the first option "Not listed? Click here."

2. In the Academic email box, enter your U-M email address (in the format: `username@umich.edu`).

Users can also create an individual account using your U-M email, but don't have to. There is a more [detailed description of access options here](#). (Unfortunately, to add to this there have been some users recently who have reported error messages when trying to login to this database. My general advice for this problem is to try using an incognito browser window and follow the steps above.)

This text is also [available for purchase](#) on the O'Reilly website.

Other Textbooks and Resources (Optional)

From time to time I may refer to examples or other content from the classic textbook *The Elements of Statistical Learning* (Second Ed.) by Trevor Hastie, Robert Tibshirani, and Jerome Friedman, published by Springer.

The entire textbook is free and available online. The second edition is here (corrected link): <https://hastie.su.domains/Papers/ESLII.pdf>

if you're an R user, you may also be interested in this text : *Introduction to Statistical Learning with R* https://web.stanford.edu/~hastie/ISLRv2_website.pdf that focuses on many of the same topics, using examples in R.

For very useful mathematical background, see the [companion webpage](#) to the book "Mathematics for Machine Learning". Copyright 2020 by Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. Published by Cambridge University Press.

Learning Outcomes

Here's a summary of some key learning objectives we have (1) for the course overall, and (2) broken down by week.

Course-wide objectives:

Understand how to correctly prepare datasets for use, e.g. feature normalization, stratified train-validate-test splits, etc.

-

Understand key families of supervised learning methods at an applied level (key parameters, decision boundary properties, etc.)

-

Understand how to evaluate and interpret results from SciKit-Learn estimators.

-

Understand how to select an appropriate supervised machine learning method for a given scenario and dataset.

-

Understand the tradeoffs inherent in different machine learning methods: speed, accuracy, complexity of hypothesis space, etc.

-

Increase awareness of issues of algorithmic bias, transparency, fairness in supervised machine learning applications.

-

Week 1:

Understand basic concepts of supervised learning

-

Apply k-nearest neighbor classification as an example of supervised learning.

-

Understand over- and under-fitting and how to detect and prevent these.

-

Week 2:

Be able to train and apply regression and classification objects (estimators) in SciKit-Learn.

-

Understand and apply linear and logistic regression, linear and kernel support vector machines

-

Use model selection methods such as cross-validation to tune the choice of model and key parameters

.

-

Week 3:

Understand and apply a wide variety of evaluation metrics to supervised learning scenarios.

-

Be able to optimize a classifier for a variety of metrics.

-

Week 4:

Learn how to apply a Naïve Bayes classifier.

-

Learn how to apply decision trees and advanced tree-based classifiers like Gradient Boosted Decision Trees.

-

Learn how to apply Neural Networks

-

- Understand what data leakage is, why it's critical to avoid, and how to detect it.

Schedule

Week 1: You'll be introduced to basic machine learning concepts, tasks, and workflow using an example classification problem based on the K-nearest neighbors method, and implemented using the SciKit-Learn library. This week's assignment has a refresher on data manipulation with numpy in part 1. Then in part 2, you work through the process of loading and examining a dataset, training a k-nearest neighbors classifier on the dataset, and then evaluating the accuracy of the classifier and using it to classify new data.

Readings (Intro to ML with Python):

All of Chapter 1

Section 2.1: Classification and Regression (short!)

Section 2.2: Over and Underfitting

Section 2.3: Introduction (short!)

Section 2.3.1: Sample datasets

Section 4.-4.2: Dummy variables and One-Hot Encoding

Week 2: We will delve into a wider variety of supervised learning methods for both classification and regression, learning about the connection between model complexity and generalization performance, the importance of proper feature scaling, and how to control model complexity by applying techniques like regularization to avoid overfitting. In addition to k-nearest neighbors, this week covers linear regression (least-squares, ridge, lasso, and polynomial regression), logistic regression, support vector machines, decision trees, and the use of cross-validation for model evaluation.

Readings:

Section 2.3.2: K-Nearest Neighbors Classification and Regression

Section 2.3.3: Linear models (Warning - long section!). This covers flavors of linear and logistic regression, and linear SVMs.

Section 2.3.7: Kernelized Support Vector Machines (SVMs).

Section 3.3: Preprocessing and Scaling

Section 4.5: Interactions and Polynomials section (for Assignment 2)

Section 5.1: Cross-Validation

Week 3: We cover evaluation and model selection methods that you can use to help understand and optimize the performance of your machine learning models. For this week's assignment, you will train a classifier to detect spam email, analyze its performance with different evaluation metrics, and then optimize the classifier's performance based on different evaluation metrics, depending on the goals of the detection task (e.g. to minimize false positives vs false negatives).

Readings:

Section 5.3.1: Evaluation introduction and goals

Section 5.3.3: Multi-class evaluation

Section 5.3.5: Using evaluation metrics in model selection

Section 5.4: Summary and outlook

Week 4: We will cover more advanced supervised learning methods that include ensembles of trees (random forests, gradient boosted trees), and neural networks (with an optional summary on deep learning). You will also learn about the critical problem of data leakage in machine learning and how to detect and avoid it. The final assignment brings everything together: you will design features for, and build your own classifier on, a prediction problem on a complex real-world dataset.

Readings:

Section 2.3.4: Naïve Bayes

Section 2.3.5: Decision Trees

Section 2.3.6: Ensembles of Decision Trees

Section 2.3.8: Neural networks

Assignments

Week 1: Review of important numpy operations on vectors and matrices. Classification with SciKit-Learn: basic concepts based on k-NN classifier. Identify likely under/overfitting scenarios.

Week 2: Regression with SciKit-Learn. Use and compare different regression methods.

Week 3: Train linear classifiers on a binary problem. Correctly normalize features and perform cross-validation. Create and interpret confusion matrices, ROC curves for a classifier. Perform grid search to find optimal parameters.

Week 4: Apply supervised learning techniques to a real-world dataset, including the methods introduced this week. Example could include a data leakage scenario. Answer questions on interpreting the results.

Quizzes

Each week will also contain a short quiz to test your knowledge of material in the lectures and readings.

Resources

Please refer to the Resources section in the Coursera shell for helpful supplemental information.

Grading and Course Checklist

I anticipate no major changes to this course grading scheme. However, as the course progresses, I reserve the right to offer additional bonus assessments or make minor adjustments/fixes as required, for any evaluation method in this course. If necessary, any such changes will always be done in a way that maximizes a student's grade across options.

Given the highly interdependent nature of MADS courses, it is essential that you properly cover all the topics in this course, each of which is important. Thus, you must "pass" ALL Assignments AND Quizzes to receive credit for this course. What does this mean? You

"pass" an assignment or quiz by getting a minimum grade of 80% regardless of whatever late penalty might also apply. If by the end of the course you have any assignments or quizzes that you did not "pass", you will be given a course grade of "incomplete" until you submit those remaining assignments or quizzes to the level of a "pass".

Example: You hand in Assignment 3 but only get a grade of 60%, which is less than 80% and thus not a "pass". You continue in the course, and to meet the above "pass" course requirement, a week later you finish the questions and achieve a score of 90% (ignoring late penalty) for Assignment 3. Thus, you now have a "pass" for Assignment 3. Your actual grade for Assignment 3 will be evaluated by the normal grading / late penalty scheme. In this case, it will be 60% (the maximum of your submission grades with late penalty included), not the 90% grade used to satisfy the "pass" criterion (which ignores the late penalty).

Course Item	Percentage of Final Grade	Passing Threshold
Week 1 Quiz	5%	80%
Week 1 Jupyter Notebook Assignments (Part 1 and 2)	15%	80%
Week 2 Quiz	5%	80%
Week 2 Jupyter Notebook Assignment	20%	80%
Week 3 Quiz	5%	80%
Week 3 Jupyter Notebook Assignment	20%	80%
Week 4 Quiz	5%	80%
Week 4 Jupyter Notebook Assignment	25%	80%

Generative AI Course Policy

This is a “foundational” course. As such, we will be working on problems that teach basic skills where it is important you do the work yourself, without code being provided for you. Thus, any assignment or quiz answers you submit must be your own (not produced by GenAI).

Feel free to use GenAI services to:

- Develop high-level pseudocode understanding of machine learning concepts and workflows.
- Review or explore concepts or ideas around the assignment questions.

Late Submission Policy

Important! Please read and understand this section, and if anything is unclear, the instructors are happy to clarify. We realize that, now more than ever, the occasional crisis might mess up your schedule enough to require a bit of extra time in completing a course assignment or quiz. Thus, we have instituted the following late policy that gives you a limited number of flexible "late day" credits.

You have a total of two (2) free late days to use during the course. One late day equals exactly one 24-hour period after the due date of the assignment (including weekends). No fractional late days: they are all or nothing. As an example, suppose you had two course late days left. Submitting any time within 24hrs of the original due date counts as using the first late day. Beyond that time, submitting any time within the next 24h counts as using the 2nd late day. After that, each additional 24h period accrues a 25% per day penalty as follows:

Once you have used up your late days, there is a 25% penalty for each subsequent 24-hour period after the deadline that an assignment is late. For example, if the due date is 11:59 p.m. ET, Monday, and you have *no* late days left, penalties would be:

Submit before 11:59 p.m. ET Tuesday: 25% deduction

Submit before 11:59 p.m. ET Wednesday: 50% deduction

Submit before 11:59 p.m. ET Thursday: 75% deduction

Submit after 11:59 p.m. ET Thursday: 100% deduction

If you were to have one late day at the end of the term the submission policy would be as follows:

Submit before 11:59 p.m. ET Wednesday: 25% deduction

Submit before 11:59 p.m. ET Thursday: 50% deduction

... and so on as above.

Important: You do not need to explain or get permission to use late days. If you choose/need to use the available late days for a week's work, you will be responsible for designating which week(s) you wish to have the late day(s) applied. Your choice could impact your final course grade due to the assignment weighting described in the Grading and Course Checklist section above. Upon completion of all assignments,

please *Slack DM the course Instructor and Lecturer(s)*(combined message) with your desired application of late days. Note that any submissions after the weekly deadline may be counted as a late submission for the week.

This late policy applies to both assignments and quizzes.

Very Important: If you intend to work on and submit an assignment after the official last day of the course (e.g. you use late days to complete a week 4 assignment past the deadline), you must make sure to start your work early enough that any assistance you might need from the instructional team can be obtained no later than the last official day of the course (i.e. stated close of the session). There is no guarantee that assistance from instructional staff will still be available after the last day of the course, since their time will already be occupied preparing for the next session's courses.

Please note: Submitting your work on time is very important in this course. Do not leave the assignments for the last minute. Ultimately, it is your responsibility to stay on track, though the instructional team may reach out to you and ask you about your progress; if you fall behind it may be difficult to catch up, and you will be at risk for not succeeding in the course. Start early, finish on time!

Important: Each week's assignments for this course are due on Mondays at 11:59 p.m. ET.

Letter Grades

The grading scale for this course will be as follows:

A+	97%
A	93%
A-	90%

B+	87%
B	83%
B-	80%
C+	77%
C	73%
C-	70%
D+	67%
D	63%
D-	60%
F	0%

Program-wide Information

Help Desk(s): How to get Help

Need help? You may reach out to UMSI or Coursera depending on the type of question you have.

- Degree program questions or general help - umsimadshelp@umich.edu
- Coursera's Technical Support (24/7) - <https://learner.coursera.help/>

Academic Integrity/Code of Conduct

Refer to the [Academic and Professional Integrity](#) section of the [UMSI Student Handbook](#). (access to Student Orientation course required).

Unless otherwise specified in an assignment, all submitted work must be your own, original work. Any excerpts, statements, or phrases from the work of others must be clearly identified as a quotation, and a proper citation provided. Any violation of the School's policy on Academic and Professional Integrity

(stated in the Master's and Doctoral Student Handbooks) will result in serious penalties, which might range from failing an assignment, to failing a course, to being expelled from the program. Violations of academic and professional integrity will be reported to UMSI Student Affairs. Consequences impacting assignment or course grades are determined by the faculty instructor; additional sanctions may be imposed by the assistant dean for academic and student affairs.

Accommodations

Refer to the [Accommodations for Students with Disabilities](#) section of the UMSI Student Handbook (access to the Student Orientation course required). Use the Student Application Form in [Accommodate](#) to begin the process of working with the University's Office of Services for Students with Disabilities.

Accessibility

Refer to the [Screen reader configuration for Jupyter Notebook Content](#) document to learn accessibility tips for Jupyter Notebooks.

Audio and Video Recording

Course Office Hours Recordings

We will be doing audio and video recording of all sessions to enable those who cannot attend Office Hours in person (virtually) on a given day to access the content. These recordings will not be made available publicly. Recordings of all sessions will be available on Coursera only to students registered for this class. As part of your participation in this course, you may be recorded should you choose to proactively participate. If you do not wish to be recorded, please do not enable your microphone or video camera during the Zoom session. Students may not copy and share the lecture videos with those not in the class, or upload them to any other online environment (this is a violation of the Federal Education Rights and Privacy Act (FERPA)). Personal recordings are prohibited except with permission. Students are prohibited from recording/distributing any class activity without written permission from the instructor,

except as necessary as part of approved accommodations for students with disabilities. Any approved recordings may only be used for the student's own private use.

Disability Statement

The University of Michigan recognizes disability as an integral part of diversity and is committed to creating an inclusive and equitable educational environment for students with disabilities. Students who are experiencing a disability-related barrier should contact [Services for Students with Disabilities](https://ssd.umich.edu/) (<https://ssd.umich.edu/>, 734-763-3000, or ssdoffice@umich.edu). For students who are connected with SSD, accommodation requests can be made in Accommodate. If you have any questions or concerns please contact your SSD Coordinator or visit SSD's Current Student webpage. SSD considers aspects of the course design, course learning objects and the individual academic and course barriers experienced by the student. Further conversation with SSD, instructors, and the student may be warranted to ensure an accessible course experience. The instructional team will treat any information that you provide in as confidential a manner as possible.

Library Access

Refer to the [U-M Library's information sheet](#) on accessing library resources from off-campus. For more information regarding library support services, please refer to the [U-M Library Resources](#) section of the UMSI Student Handbook (access to the Student Orientation course required).

Student Mental Health

University Students may experience stressors that can impact both their academic experience and their personal well-being. These may include academic pressures and challenges associated with

relationships, mental health, alcohol or other drugs, identities, finances, etc. If you are experiencing

concerns, seeking help is a courageous thing to do for yourself and those who care about you. If the

source of your stressors is academic, please contact me so that we can find solutions together. Ashley

Evearitt, a Counseling and Psychological Services (CAPS) counselor, is embedded in UMSI, schedule an

appointment with her by copying and pasting the following URL into your browser:

<https://bit.ly/Evearitt-CAPS-UMSI>

For personal concerns, U-M offers a variety of resources, many which are listed on the [Resources for](#)

[Student Well-being](#) webpage. You can also search the University's [Resources for Stress and Mental Health website](#) for additional resource listings..

Student Services

Refer to the [Introduction to UMSI Student Life](#) section of the UMSI Student Handbook (access to the Student Orientation course required).

Technology Tips

We will be using Slack, Zoom, Google Docs, and Google Sheets to facilitate communication. Your own work on the project will be done in Jupyter.

We have created a Jupyter environment for you that is functionally equivalent to SIADS 516, which is a superset of the base MADS environment. You can access that environment via the "ungraded lab assignment" in Coursera. You can use that environment or choose to use any of the environments from courses you have already completed. Alternatively, you can use your own locally installed environment. Another possibility is to use [Google Colaboratory](#), which may facilitate collaboration.

Working Offline

While the Coursera platform has an integrated Jupyter Notebook system, you can work offline on your own computer by installing Python 3.8.10 and the Jupyter software packages. For more details, consult the [Jupyter Notebook FAQ](#).