Applied Machine Learning Course

Instructor Guide

I. Introduction

II. Getting Started

III. Instruction

IV. Additional Supporting Documents

I. Introduction

A. How to Use This Instructor Guide

This instructor guide should be used by the instructional team to prepare to teach the applied machine learning course to university students throughout an academic semester, quarter, or summer.

This **introduction** section provides an overview of the course content, goals, and proposed outcomes.

Getting started includes suggestions for the instructional team and student prerequisites, as well as the materials required for a successful course.

Instruction provides both a general and in-depth look at the course content, from a broad overview to the tracks, units, and lectures, and learning activities that form the curriculum. Next, information about the program's capstone project can be found here.

Additional supporting documents links to all other relevant documents required for teaching this applied machine learning course.

B. Course Overview

As machine learning (ML) becomes a powerful tool across industries — from healthcare and retail to investment banking and insurance — there is a growing need for a workforce that understands how to apply ML strategically and use ML models to collaborate with data scientists and engineers for maximum business impact.

In this course, students will learn the fundamentals of machine learning (ML) to prepare for a role at the intersection of data science, computer science, and the individual student's field of study or interest. Students will be immersed in different ML tools and models, develop skills to assess when ML is the right solution to a given problem, learn how to prepare and identify issues with data, and hone their coding skills in Python.

This curriculum includes the following topics:

- data investigation, cleaning, transformation, analysis, and visualization
- ML techniques like classification, regression, and clustering
- bias, responsible AI, and machine learning fairness

Students will work in project-based teams dedicated to exploring and solving data problems based on their own professional aspirations. To complement the real-world technical content, students will also learn professional skills essential to entering and succeeding in today's workforce: professional networking, giving and receiving feedback, presenting with confidence, and navigating the recruiting process.

C. Student Learning Outcomes

Throughout this course, students will reach the following learning outcomes:

1	Investigate, clean, and visualize data.
2	Understand and frame a problem as a supervised machine learning problem including whether it is a regression or classification problem and to incorporate the application requirements.
3	Apply and tune common machine learning models (for regression and classification) in Python by making use of multiple ML toolkits.
4	Demonstrate the ability to qualitatively and quantitatively evaluate the quality of trained regression and classification models.
5	Communicate technical concepts (both orally and in writing) for an audience who may have a limited technical background.
6	Identify the potential for bias in ML models and explain its implications along with techniques for mitigating it.

II. Getting Started

A. The Instructional Team

The first step toward developing a successful program is to identify and build a team of instructors and program staff. It is recommended to have at least 2-3 program staff in the classroom each day, including a technical instructor and two teaching assistants.

B. Student Prerequisites

Prior to beginning course content, students are **required** to have:

- completed at least one course in computer science and a second course in some technical subject area
- familiarity with Python
- familiarity with Straight Line Equation
- familiarity with Functions

In addition to those required topics, it is **preferred** that students have:

- familiarity with Matrix Algebra
- familiarity with Normal Distribution
- familiarity with Hypothesis Testing
- familiarity with P-Values For Data Scientists

- familiarity with [Scientific Notation]

In addition to the above required and preferred topics, it is **optional** that students have:

- familiarity with [Binomial Distribution]
- familiarity with [Logs]

III. Instruction

A. Curriculum Overview

The curriculum is organized into seven learning **tracks,** each of which consists of a sequenced series of **units.** These units, each with a lecture and **lab** or other **learning activity**, prepare students to complete an independent or group mini project as part of each track and then a **capstone project** at the end of the course.

The tracks and mini projects will make up half or slightly more than half of the program. After that, students' time will be dedicated to completing the capstone project.

In addition to the seven learning tracks (T01 - T07), the curriculum also includes two optional tracks: miscellaneous technical topics (TXX) and career development curriculum (TCD). The TXX track can be utilized for one-off technical lessons during capstone project weeks. The TCD track is designed to be interwoven throughout the curriculum starting in week 1.

B. Instructional Methodology

This applied machine learning course was developed by software engineers and university professors to prepare university students to apply data science principles across a variety of fields. To that end, care has been taken to ensure the content, instructional methods, and assignments are relevant to the current state of data science and machine learning as they are practiced in industry. Furthermore, content focuses more on the application of data science and machine learning and less on the mathematical and algorithmic details of the algorithms.

Tracks are organized to connect related concepts and technologies. It is suggested they be taught in order. The units within the tracks are arranged so new information is presented via brief lectures and reinforced through learning activities like labs and mini projects. A lecture and either a lab or other learning activity make up a single unit within a track. To mimic work flows students are likely to experience in the industry, a mix of individual and group-based work is utilized within each track.

Students will complete five mini projects during the first part of the program. The last part of the program is almost solely devoted to the capstone project. This culminating project is prioritized for several reasons. First, the project provides students an opportunity to integrate their learning from the course to solve a particular problem or meet a specific need. Students integrate technical concepts, project management, organizational skills, and ethical considerations to create a meaningful final product. Second, data science employers look to see projects applicants have undertaken, the skills they built while working on them, and what they learned from the experience. The capstone project provides students with a meaningful work-like experience to discuss in interviews. Students present these projects on the last day of the program, followed by a “graduation” ceremony.

C. Curriculum Scope & Sequence

a. **Tracks:** Tracks reflect the seven major topics of this course, beginning with *T01: What is Machine Learning?* and ending with *T07: Capstone Project*. In addition to the seven main tracks, there are two additional tracks: miscellaneous technical topics (TXX) and career development (TCD).

T#	Track
00	Prerequisites
01	What is Machine Learning?
02	Data
03	Regression
04	Classification
05	Deep Learning
06	Other Models
07	Capstone Project
XX	Miscellaneous Technical Topics
CD	Career Development

b. **Units:** Units are typically made up of one lecture and one ab and/or activity. An example of a track, *T01: Introduction to Machine Learning*, and its units are below.

T01: What is Machine Learning?	
T01-00	Introduction to Machine Learning
T01-01	Introduction to Machine Learning Models
T01-02	Machine Learning Fairness

c. **Learning Activities:** Learning activities such as labs and group activities are used to practice and reinforce what was taught in the corresponding unit’s lecture.

d. **Mini Projects**: Mini projects cover a range of topics in Track 02 through Track 06. Track 07 is the capstone project.

D. Full Curriculum

a. **Technical Curriculum**

Unit #	Unit Name	Unit Objectives
Track 00: Prerequisites		
T00-00	Introduction to Python	Create, use, and troubleshoot variables Read and write Python statements, expressions, conditionals, loops, and functions
T00-01	Intermediate Python	Build a basic Python object Build a hierarchy of objects Distinguish procedural from object-oriented programming styles Distinguish a class from an instance Interpret different types of exceptions Create your own exception class Define functions using lambda syntax Interpret list comprehension notation Create lists using list comprehension with for and if statements
Track 01: What is Machine Learning?		
T01-00	Intro to Machine Learning	Identify and use basic machine learning terminology.
T01-01	Intro to ML Models	Distinguish between different types of ML models.
T01-02	ML Fairness	Identify ways in which ML biases can have real ethical consequences. Article #1: The Reel Truth: Women Aren't Seen or Heard Article #2: Color Film Was Designed to Take Pictures of White People, Not People of Color: The Unfortunate History of Racial Bias in Photography (1940-1990) Article #3: Machine Bias There's software used across the country to predict future criminals. And it's biased against blacks. Article #4: The AI Text Generator That's Too Dangerous to Make Public
Track 02: Data Analysis & Manipulation		
T02-00	Introduction to Colab	Identify where Colab fits in the development

		environment space Edit markdown in a notebook Edit and run code in a notebook
T02-01	Introduction to Pandas	Create, analyze, and modify a Pandas Series Create, analyze, and modify a Pandas DataFrame
T02-02	Intermediate Pandas	Apply filters to Pandas DataFrames Group data contained in Pandas DataFrames Merge data across multiple Pandas DataFrames Sort data contained in Pandas DataFrames
T02-03	Visualizations	Create and interpret pie charts Create and interpret bar charts Create and interpret line charts Create and interpret scatter plots Create and interpret heat maps Create charts with Matplotlib Create charts with seaborn Determine which visualization is most appropriate for a dataset
T02-04	Acquiring Data	Upload data to Colab Download data from public URLs Download and obtain data from Kaggle Unzip compressed data
T02-05	Exploratory Data Analysis	Identify and calculate statistics for a DataFrame Analyze data across DataFrame objects Select appropriate visualizations to use for analysis Interpret visualizations to answer questions about a dataset Identify and fill in missing data points in a dataset Identify and correct broken data points in a dataset

T02-06	Data Processing Project **[Project #1]**	Acquire and load dataset(s) into Pandas structures Inspect data columns description and statistics Explore data to understand relationship between features Use visualizations to convey trends
T03: Regression		
T03-00	Introduction to Regression No Colab for this lesson. For extra practice manually calculating L1 and L2 loss, see xx_misc Loss Functions.	Identify the components of a linear regression model Identify how the machine learning process applies to linear regression Distinguish between parameters and hyperparameters
T03-01	Introduction to scikit-learn	Load sample data packaged with scikit-learn Generate sample data using scikit-learn Transform data using scikit-learn Train a simple model and make predictions using that model Create a data-processing and model-training pipeline Create metrics around model performance Visualize predictions returned from a model
T03-02	Linear Regression with scikit-learn	Train a linear regression model using scikit-learn on real data Use root mean squared error (RMSE) to evaluate a linear regression model Visualize features, targets, and predicted targets using a scatter plot (Optional) Create a linear regression model using the normal equation and optimize using the pseudoinverse (Optional) Practice stochastic gradient descent and mini batching
T03-03	Regression Quality	Extract quantitative measurements of a regression model's predictions Perform qualitative judgments of a regression model's predictions
T03-04	Polynomial Regression	Apply polynomial models to regression problems

		Recognize when a model might be overfitting Correct overfitting using techniques such as Lasso, Ridge, and ElasticNet regularization
T03-05	Introduction to TensorFlow	Distinguish between types of tensors (scalars, vectors, matrices, cubes, etc.) Identify key differences between TensorFlow 1 and TensorFlow 2 Perform basic linear algebra operations on tensors using TensorFlow Convert tensors to NumPy arrays and Python lists
T03-06	Linear Regression with TensorFlow	Use the TensorFlow Estimator API to build a model Adjust model hyperparameters Interpret model performance metrics
T03-07	Neural Networks No Colab for this lesson.	Identify the elements of a perceptron Adjust weights and bias in a neural network Track a basic neural network prediction through hidden layers and activation functions
T03-08	Regression with TensorFlow	Use the TensorFlow/Keras API to build a deep neural network Understand the implications of activation function choice
T03-09	Regression Project **[Project #2]**	Argue the merits (or lack thereof) for a regression model Discuss the ethics of a regression model Explore a dataset with minimal guidance Build a regression model and perform hyperparameter tuning Judge the quality of a regression model
T04: Classification		
T04-00	Intro to Classification No Colab for this lesson.	Differentiate between classification and regression Interpret accuracy, precision, recall, and F1 scoring to classification models
T04-01	Binary Classification	Create a logistic regression model for a binary classification problem

		Interpret a confusion matrix for a binary classification model Use a grid search to find optimal hyperparameters for a model
T04-02	Multiclass Classification	Build a classification model for data with more than two classes Use cross-validation to evaluate a model trained with a small amount of data Create a model pipeline for training and predicting
T04-03	Classification with TensorFlow	Create a classification model with TensorFlow Use a trained TensorFlow model to make classification predictions
T04-04	Classification Project **[Project #3]**	Define, build, train and evaluate a Linear Classifier model in TensorFlow Submit predictions to a Kaggle challenge
T04-05	Introduction to Image Classification	Utilize effective strategies for feature reduction in image classification Perform multiclass image classification using a deep neural network Prevent overfitting using early stopping and dropout
T04-06	Images and Video	Resize, pad, and change the orientation of an image Load an image with OpenCV Change the color encoding of an image Modify the size, cropping, and orientation of an image Use OpenCV to process video
T04-07	Saving and Loading Models	Implement the process to save the state of a model Revive and use a persisted model
T04-08	Video Processing Project **[Project #4]**	Use OpenCV to process images and video Use a pre-trained model to identify and label objects in each frame of a video Judge the classification quality and when to apply predicted labels
T04-XX	Classification Gone Wrong	Identify examples of classification models that had unintended, harmful effects

	No Colab for this lesson.	Discuss potential causes of bias and harmful errors in classification Discuss ways to mitigate bias
T05: Deep Learning		
T05-00	Convolutional Neural Networks	Identify the components of a CNN Identify the effect of different filters
T05-01	Recurrent Neural Networks	Use TensorFlow to build a recurrent neural network Feed time series data to a neural network to make sequence predictions
T05-02	Natural Language Processing	Utilize text processing and feature extraction tools Train NLP models using bag-of-words and sequential representations
T05-03	Autoencoders	Understand the fundamental structure of autoencoders Implement an autoencoder for compressing and denoising images Combine multiple models using a wrapper model
T05-04	Transfer Learning	Become familiar with the PyTorch API Employ the fastai API to implement a CNN
T05-05	Image Classification Project **[Project #5]**	Discuss ethical implications of a model that involves medical decisions Create a classification model end-to-end, including parameter tuning and final validation
T06: Other Models		
T06-00	Clustering No Colab for this lesson. This unit requires materials - an assortment of fasteners (or screws, bolts, pins, buttons, etc.).	Differentiate clustering from regression and classification Manually cluster objects using a tactic similar to the k-means algorithm
T06-01	k-Means	Identify the difference between supervised and unsupervised learning Create a k-means model Interpret the output of a k-means model

T06-02	Embeddings	Describe embeddings, why they are used, and how they are trained Implement an embedding in practice
T06-03	Decision Trees & Random Forests	Create and apply a decision tree algorithm for classification Perform ensemble learning using random forests Apply limits to depth and split size to reduce overfitting
T06-04	K Nearest Neighbors	Describe the basic concept of KNN Use KNN to solve a classification problem
T06-05	Support Vector Machines	Define problems for which support vector machines are a good fit Understand the primary settings used to tune a support vector machine and their tradeoffs
T06-06	Bayesian Modeling This lesson should come *after* Bayes' theorem is introduced in TXX-05 Probability and Statistics.	Identify and describe the components of Bayes' Theorem Predict spam or ham using naive Bayes Predict review sentiment (+ or -) using naive Baye
T06-07	XG Boost	Understand the idea of gradient boosting Implement the XGBoost algorithm
T07: Capstone Project		
T07-00	Capstone Project Introduction No deliverable for this lesson.	Identify capstone project basics and graded elements
T07-01	Design Documents	Create design documents for an independent project
T07-02	Project Ideation Materials needed: Dot-stickers Self stick easel pads (flipchart) Colorful sharpies Lots of post-it notes	Identify ideas for a capstone project that are appropriate and scalable to fit the time and resource constraints

b. **Miscellaneous Technical Topics**

TXX: Miscellaneous Technical Topics		
TXX-00	Activation Functions	Identify different types of activation functions Identify where activation functions are used Visualize activation functions
TXX-01	Big O This lesson is especially helpful for students engaging in technical interviews for software engineering roles.	Analyze the time and space complexity of simple pieces of code Identify the most common running times, and order them from slowest to fastest
TXX-02	Dimensionality Reduction Lab has no exercises/deliverables.	Identify when and when not to use PCA Select an optimal number of components for PCA Calculate explained variance
TXX-03	History of ML No Colab for this lesson.	Engage in a trivia game that covers some of the major names and events in machine learning's history
TXX-04	Loss Functions Not a stand-alone lesson. Additional practice on loss functions. Intended to be an add-on to T03_00 Intro to Regression.	Manually calculate L1 and L2 loss
TXX-05	Probability and Statistics The lab for this lesson is intended to be used as a glossary rather than a typical lab.	Calculate the probability of an event from a probability distribution Calculate expected value Describe basic statistical sampling and generate a representative sample from a dataset Calculate measures of center and measures of spread
TXX-06	Regular Expressions	Identify and explain regular expressions Use regular expressions with Python

c. Career Development Curriculum

TCD: Career Development		
Unit #	**Session**	**Objectives**
TCD-01	Life Paths	Get to know one another through sharing

		important events and/or time periods in a "life path" model
TCD-02	Resume Workshop	Employ best practices for a technical resume
TCD-03	The Art of Networking	Employ best practices for meeting and engaging with new professional contacts
TCD-04	Giving & Receiving Feedback	Engage with and apply tips for giving and receiving feedback in a professional environment
TCD-05	Communicating with Recruiters	Discuss the role technical recruiters play in the application and interview process
TCD-06	Preparing for Interviews	Engage with basic tips for both technical and behavioral interviews
TCD-07	Presenting with Confidence	Employ best practices for delivering strong presentations with confidence

E. Instructional Resources

All course content is housed in Github.

Lecture slides have been developed in Markdown. These slides contain lesson content as well as detailed speaker notes for every unit. They have been converted to PowerPoint for use by individual classrooms using the marp tool. Supporting docs for these units have also been developed in ODT and PDF formats. These materials are modifiable, so individual instructors can adapt the materials to their teaching styles and add individual insights.

Labs are IPython notebooks targeted at the Colab environment and compatible with most Jupyter environments. Each Colab contains student, instructor, and content data. The instructor versions of the labs are in a locked zip file in each unit.

The course will primarily use scikit-learn and TensorFlow v2. Other toolkits can be used as long as they are easy to install on Colab.

F. Grading

How grades are assigned for this course is left up to the sponsoring organization and/or universities. If grades are assigned, the sponsoring organization/university should work together to determine the grading scale and point values, and a university-level instructor should formally assign students grades.

G. Suggested Schedule

Below are suggestions for planning and executing a full-time, campus-based course based on a previous iteration of the program. Course instructors should adapt these guidelines based on the amount of time allotted for their particular course.

a. Sample Day

Time	Activity
9:30-10:00am	Daily welcome, energizer, review of yesterday's material, Q&A
10:00-10:30am	Unit lecture
10:30-12:00pm	Unit Colab
12:00-1:00pm	Lunch
1:00-2:00pm	Unit lecture
2:00-2:15pm	Afternoon energizer
2:15-4:00pm	Unit Colab
4:00-4:30pm	Debrief the day, Q&A, forecast tomorrow

b. **Week-by-Week Schedule**

The first iteration of this course was a full-time, on-campus program. The following schedule will be useful for planning a similar campus-based course.

<table>
<tr><th>8+ weeks prior to students' arrival</th></tr>
<tr><td>
- Hire all program staff, including teaching assistants.
- Confirm lodging for students, if applicable, and ensure accessibility.
- Reserve classroom space(s) and ensure accessibility.
- Set up campus-based meal plans or alternative for students.
- Begin reviewing curriculum and choosing units/supplemental materials.
- Email students to communicate the following:
    - Confirm acceptance and ask for program commitment
    - Program details that allow students to book travel and give a basic overview of course
    - If program will be residential, include details for getting to campus and living in residence halls
</td></tr>
<tr><th>4 weeks prior to students' arrival</th></tr>
<tr><td>
- Check in with all relevant campus partners (dining, residence life, student services).
- Confirm how students will receive mail, if applicable.
- Begin meeting as an instructional team weekly in preparation for students' arrival, delegating instructional tasks and reviewing and personalizing curriculum.
</td></tr>
</table>

- Review course materials and purchase necessary items (chart paper, white board markers, career development materials, etc).
- Email students to communicate the following:
    - Specific arrival instructions
    - Contact information for program staff
    - Any required prep work

2 weeks prior to students' arrival

- Meet with relevant campus partners virtually or in person about all arrival, check-in, and programmatic details related to room and board.
- Plan for students to have any special requests or accessibility requests routed to the correct offices for approval.
- If you'll be organizing guests speakers to visit the course, begin requesting and confirming visits.
- Email students to communicate the following:
    - Introduce instructors to students
    - Student handbook that must be read and signed
    - Course syllabus and schedule
    - Information on their mailing address during program

1 week prior to students' arrival

- Review career development curriculum.
- Ensure roles of instructional team members are clear.
- Decide on programming for week one to ensure students get to know each other and build psychological safety.
- Email students to communicate the following:
    - Reiterate arrival instructions
    - Remind students about any required materials they should bring or prep work that should be completed prior to day 1

First week of the course
- Day one should consist of an overview of the course as a whole and daily structure of class. - Include one longer ice breaker activity for students to learn one another's names. Ice breakers should continue each day within the first week so students learn one another's names and also build psychological safety. - Life Paths is a great Career Development activity for day one or two. - Take a group picture.
Ongoing throughout the course
- Prior to the start of each day, add the agenda to a whiteboard or have it available in a presentation slide to give students a sense of the day's work. - Implement a system for due date reminders.

- Send out a daily exit survey to students to get a sense of the pacing of the program and any outstanding questions. - At the end of each day, hold a debrief with instructional staff. This time can be used to ask what went well, what didn't go well, and what should be changed in the upcoming day or week. The abovementioned exit survey can be used to guide this conversation.
Halfway through the course
- Meet as an instructional team about capstone project team selection and determine best course of action. - Instruct on specific Program Management principles, in advance of capstone projects so that students know tools at their disposal for self-management. - Begin planning graduation/end-of-program celebration (normally takes place on the last day of class).
Second to last week of the course
- Plan an "exit survey" for students to complete on the last day of the course to give feedback on the course. - Go over presentation requirements so all students are clear on the parameters of the presentation and can begin planning accordingly - they should now shift to work on presentations as well as finishing up their projects. - If students will be inviting external guests as audience members to project presentations, you can offer a template invitation with details.
Last week of the course
- Toward the beginning of the week, have students sign up for practice presentation spots with members of the instructional staff with enough time to receive and incorporate feedback before the final presentation.

H. Capstone Project

Included in this section are suggestions for capstone project parameters, timeline, student deliverables, grading, and final presentations.

a. **Overview:** The capstone project starts roughly halfway through the course and is meant to be completed in a group collaboration with the following learning objectives:
   - Apply technical ML concepts and data skills from the first half of the course on a sizeable dataset and challenging problem
   - Demonstrate professional development skills, such as working in a group with different personalities, giving and receiving feedback, communicating results, and project management
   - Create and facilitate a recorded demo or presentation of capstone project work

b. **Student Deliverables:** Students will complete each of the following deliverables for their total capstone. The percent value next to each item reflects how much effort and time are likely to be spent on that particular assignment and can therefore inform assessment.

- **Design doc - 20%**
    - Goal(s) for the project - ideally long term, and realistic for the project duration
    - Who will play what roles (program manager, note taker) during each project phase
    - Describe the dataset(s), data acquisition, and data preparation
    - Explain problem space and motivated questions
    - Approach and list of tasks
- **Ethical consideration worksheet - 10%**
- **Notebook - 50%**
    - The notebook should read like a formal report and follow a linear flow from start to finish with both narrative and code blocks.
    - The notebook should be internally complete in the sense that a "reader" should understand the motivating question, goals, dataset, model, and ethical considerations.
    - Markdown cells with narrative should precede and follow each code block.
    - For each piece of code, describe the purpose, an overview of how it works, and how to interpret results.
    - Discuss current limitations and future improvements.
- **Project demo or presentation - 20%**
    - Overview of the project and the problem it seeks to solve
    - Conclusions and findings
    - Things that went well, things that did not go well, and lessons learned
    - Next steps

c. **Recommendations for Final Presentations:** Final presentations should occur within the final day or two of the course. If possible, encourage students to invite guests like faculty, family, or friends. Record the presentations so students have it as an artifact of their capstone project work.

- Each group's project demo or presentation should adhere to the following format and cover all the points here:
    - Overview of the project and the problem it seeks to solve, if applicable
    - Conclusions and findings
    - Things that went well, things that did not go well, and lessons learned
    - Next steps, if applicable
- Presentations should be 15 minutes each plus 5 minutes for questions. Each member of the team should present for roughly the same amount of time.
- Presentations should be visually interesting but also follow best practices for verbal presentation skills. Namely, it is important for all points to be conveyed clearly.
- Provide opportunities for groups to practice presenting for the instructional team during the 2-3 days leading up to final presentations, allowing sufficient time to incorporate feedback and refine their presentations.

- During the final presentations, identify one person to time them and hold cards indicating when 5, 3 and 1 minutes remain. This will help presentations stay on schedule and support students to stay within their 15-minute window.

d. **Project Timeline:** Each phase can be condensed into a few days or expanded into a week or more. Keep in mind that at this point, students should be taking on more responsibility and should only receive this roadmap as a guideline. They should ultimately be the owners of weekly outcomes. Of course, instructors should adjust how much guidance their individual students need.

Project Phase 1	• Form teams • Explore and pick project topics • Write a design doc on how you approach building the ML model • Create a high level project plan of who's doing what tasks by when
Project Phase 2	• Acquire & prep dataset • Start defining and training ML model • Beginning project presentation: discuss acquiring data, exploratory data analysis, and justification for first model
Project Phase 3	• Use data to tune model and validate • Mid project presentation: share iterations of model tuning with a focus on challenges and questions, welcome feedback from peers and instructors • Group and individual meetings with the instructional team
Project Phase 4	• Final iterations on model predictions • Review results and prepare conclusions • Practice final presentation and demo, collect, and incorporate final feedback

e. **Recommendations for Group Management**

- **Roles:** To ensure a robust project experience, students should take turns working in different capacities. Faculty are encouraged to clarify this expectation and explain the rationale up front. Each team's design doc should include a schedule of when these roles will rotate and to whom throughout the project phases:
    - **Program Manager:** Each student should take a turn in the role of "program manager," or the person who is purposefully directing the team and assigning action items. This role builds useful professional skills, and rotating ensures no one person gets stuck with all of the administrative tasks to keep the team running.
    - **Note Taker:** Each student should be responsible for maintaining a "decision log" during one phase of the project. Entries in decision logs

may include why the team went forward with one thing over another, how a problem got resolved, unresolved tasks or questions, etc. Decision logs will be housed in a common repo and accessible to students and instructional staff.

- **Project Plans:** Groups should set up their own project plan at the outset of capstone project work. In order for students to organize themselves and their work, they should not be given a precise template. Students' project plans do not have to be elaborate. The best project plans ensure tasks assigned to an owner with a clear timeline for delivery and perhaps align to greater project milestones.

- **Daily Standups:** It is helpful to have a morning and afternoon "standup." In the morning each group takes 10 minutes or less to share how they will prioritize their time for the day and what they feasibly will get accomplished. This is also a good time for group members to raise issues or anything "blocking" them that they could use the group's help to mitigate. Then at the end of the day, groups should come back together as a class to share what they accomplished during the day, anything they needed help with, and what their focus will be the following day. This bookend standup structure encourages students to make good use of class time and helps ensure students don't get stuck for long periods of time.

f. **Daily Project Schedule**
- Morning team stand-ups
- Project work (multi-hour block)
- (Optional) ad hoc topic session (at most 2-3 times per phase)
- Lunch
- Project work (multi-hour block)
- (Optional) professional skills session or guest speaker (at most 1 per phase)
- End of day team meetings and stand-ups

IV. Additional Supporting Documents

A. Supplemental Course Materials

This list of resources includes brief learning activities and resources that do not warrant a full lecture and colab for this course, but they are included to help fill class time as needed.

Supplemental Course Materials	
AI/ML Critical Reading List List of Reading Links	This is a useful compendium of resources of machine learning principles and AI ethics.
Big Query and Pandas	This is the canonical source on downloading BigQuery datasets into pandas dataframes.

Downloading BigQuery data to pandas using the BigQuery Storage API	
Big Query and Jupyter Visualizing BigQuery data in a Jupyter notebook	This is the canonical source on visualizing BigQuery data in a Jupyter notebook.
Intro to SQL W3Schools SQL Tutorial	This is a good and simple introduction to SQL.