

Machine Learning Engineering Career Track

Capstone Project

Learning Objective

- Complete a capstone project with sufficient breadth and depth to demonstrate all of the machine learning engineering skills presented in the course, including skills in programming, machine learning, deep learning, and deployment of production applications.
- Complete and present a cohesive project that is ready to be utilized as the core part of a professional portfolio.

Criteria	Meets Expectations
Phase 1 of Capstone: Building a working prototype	
Time Estimate	60 hours
Completion	<p>The submission includes all of the following core components uploaded to Github:</p> <ul style="list-style-type: none"><input type="checkbox"/> Code for the project.<input type="checkbox"/> The Github submission is complete, with all the different parts of phase 1 of the capstone project neatly organized in one repository. The repository is well-documented, with a clear README page.<input type="checkbox"/> All intermediate capstone project submissions.<ul style="list-style-type: none">• Step 1: Initial Project Ideas (No rubric)• Step 2: Project Proposal (Rubric)• Step 3: Data Collection (Rubric)• Step 4: Data Wrangling (Rubric)• Step 5: Machine Learning / Deep Learning Prototype (Rubric)• Step 6: Scale Your Prototype (Rubric)

Understanding & Process	<p>The submission demonstrates an understanding of and progress in core project objectives:</p> <ul style="list-style-type: none"> ❑ A problem was selected that has practical application, the value for a client is justified, and the ways in which the outcomes can be used are described; the project was scoped appropriately for the course. ❑ The data utilized in the project shows understanding of how to acquire, wrangle, and clean data. Datasets were well-chosen and relevant to the problem. ❑ The technical approach used for the problem and data with a demonstrated understanding of how to: <ul style="list-style-type: none"> • Select and utilize appropriate algorithms and applications. • Apply and justify utilized machine learning and deep learning techniques (as applicable to the project), including feature selection and evaluation metric/technique. ❑ Write code that is clear, understandable, and well-documented so that it can be easily deployed to production in phase 2 of the project.
-------------------------	--

Criteria	Meets Expectations
Phase 2 of Capstone: Deploy to Production	
Time Estimate	40 hours
Completion	<p>The submission includes all of the following core components uploaded to Github:</p> <ul style="list-style-type: none"> ❑ Code for the project. The Github submission is complete, with all the different parts of the capstone project neatly organized in one repository. The repository is well-documented, with a clear README page.

	<ul style="list-style-type: none"> ❑ A running application that can be used via a simple user interface. Instructions to access the application are provided as part of the README in Github. The application doesn't need to run 24/7, but can be spun up by the student when requested. ❑ All intermediate capstone project submissions. <ul style="list-style-type: none"> • Step 1: Deployment architecture (Rubric) • Step 2: End-to-end code with testing (Rubric) • Step 3: Deployment implementation (Rubric)
Understanding & Process	<p>The submission demonstrates an understanding of and progress in core project objectives:</p> <ul style="list-style-type: none"> ❑ The student proposed a deployment architecture that matches the scale and nature of the problem, addresses concerns about how the system will be deployed, monitored, debugged and how it will perform in production ❑ The student shows an understanding of various tradeoffs involved with deployment tools and decisions. ❑ The code is clear and well-documented, runs in a self-contained manner, and is adequately tested using best practices. ❑ The technical approach used for the deployment with a demonstrated understanding of how to: <ul style="list-style-type: none"> • Implement a deployed system based on a proposed architecture plan, including data pipelines, logging, and monitoring. • Design, implement, test and document an API for a real application • Create a simple user interface using a tool such as Flask to let others use the application

