UDACITY

NANODEGREE PROGRAM SYLLABUS

Machine Learning DevOps Engineer

Overview

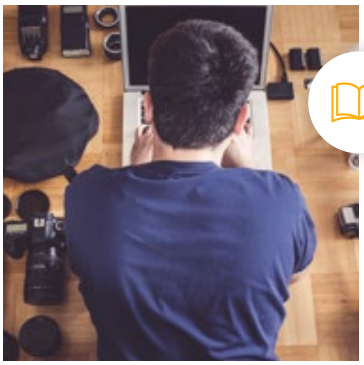
The Machine Learning DevOps Engineer Nanodegree program focuses on the software engineering fundamentals needed to successfully streamline the deployment of data and machine-learning models in a production-level environment. Students will build the DevOps skills required to automate the various aspects and stages of machine learning model building and monitoring over time.

Educational Objectives:

Students who graduate from the program will be able to:

- Implement production-ready Python code/processes for deploying ML models outside of cloud-based environments facilitated by tools such as AWS SageMaker, Azure ML, etc.
- Engineer automated data workflows that perform continuous training (CT) and model validation within a CI/CD pipeline based on updated data versioning
- Create multi-step pipelines that automatically retrain and deploy models after data updates
- Track model summary statistics and monitor model online performance over time to prevent model-degradation

Estimated Time: 4 Months



Prerequisites: Prior experience with Python and Machine Learning

Flexible Learning: Self-paced, so you can learn on the schedule that works best for you.

Need Help?
udacity.com/advisor
Discuss this program with an enrollment advisor.

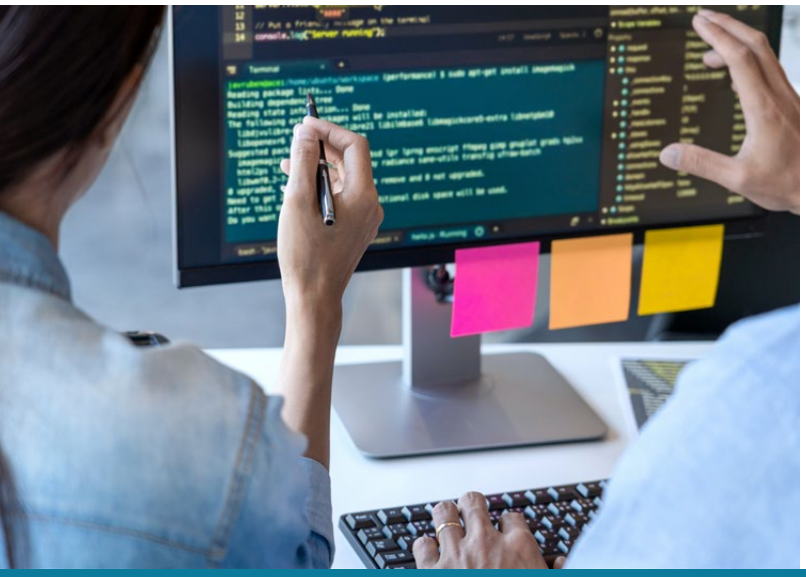
*The length of this program is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. If you spend about 5–10 hours per week working through the program, you should finish within the time provided. Actual hours may vary.

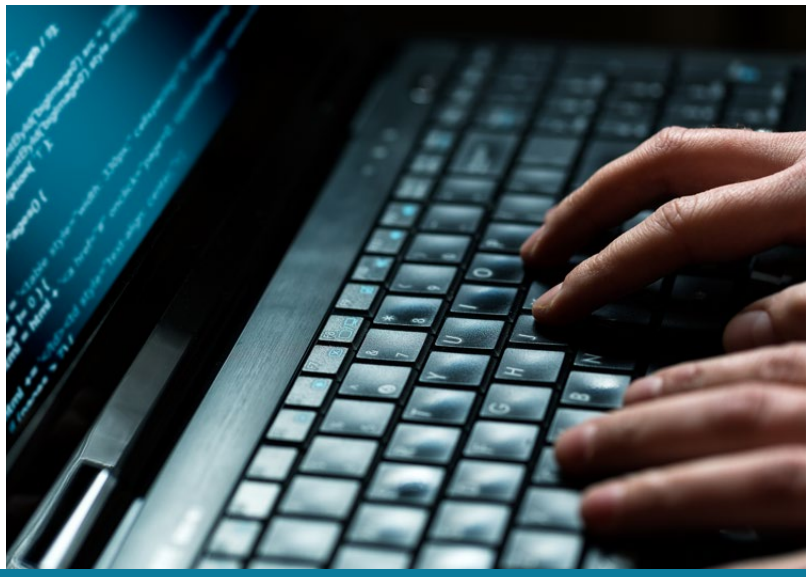
Course 1: Clean Code Principles

Develop skills that are essential for deploying production machine learning models. First, you will put your coding best practices on autopilot by learning how to use PyLint and AutoPEP8. Then you will further expand your Git and Github skills to work with teams. Finally, you will learn best practices associated with testing and logging used in production settings to ensure your models can stand the test of time.

Course Project :
Predict Customer Churn with Clean Code

In this project, you will implement your learnings to identify credit card customers most likely to churn. The completed project will include a Python package for a machine learning project that follows coding (PEP8) and engineering best practices for implementing software (modular, documented and tested). The package will also have the flexibility to run interactively or from the command-line interface (CLI). This project will give you practice using your skills for testing, logging and coding best practices from the lessons. It will also introduce you to a problem data scientists across companies face all the time: How do we identify (and later intervene with) customers who are likely to churn?





		LEARNING OUTCOMES
LESSON ONE	**Coding Best Practices**	• Write clean, modular and well-documented code • Refactor code for efficiency • Follow PEP8 Standards • Automate use of PEP8 standards using PyLint and AutoPEP8
LESSON TWO	**Working with Others Using Version Control**	• Work independently using git and Github • Work with teams using git and Github • Create branches for isolating changes in git and Github • Open pull requests for making changes to production code • Conduct and receive code reviews using best practices
LESSON THREE	**Production Ready Code**	• Correctly use try-except blocks to identify errors • Create unit tests to test programs • Track actions and results of processes with logging • Identify model drift and when automated or non-automated retraining should be used to make model updates

Course 2: Building a Reproducible Model Workflow

This course empowers the students to be more efficient, effective and productive in modern, real-world ML projects by adopting best practices around reproducible workflows. In particular, it teaches the fundamentals of MLOps and how to: a) create a clean, organized, reproducible, end-to-end machine learning pipeline from scratch using MLflow b) clean and validate the data using pytest c) track experiments, code and results using GitHub and Weights & Biases d) select the best-performing model for production and e) deploy a model using MLflow. Along the way, it also touches on other technologies like Kubernetes, Kubeflow, and Great Expectations and how they relate to the content of the class.

Course Project :
Build an ML Pipeline for Short-term Rental Prices in NYC

Students will write a Machine Learning Pipeline to solve the following problem: a property management company is renting rooms and properties in New York for short periods on various rental platforms. They need to estimate the typical price for a given property based on the price of similar properties. The company receives new data in bulk every week, so the model needs to be retrained with the same cadence, necessitating a reusable pipeline. The students will write an end-to-end pipeline covering data fetching, validation, segregation, train and validation, test, and release. They will run it on an initial data sample, then re-run it on a new data sample simulating a new data delivery.

	LEARNING OUTCOMES	
LESSON ONE	**Machine Learning Pipelines**	• MLOps fundamentals • Version data and artifacts • Write a ML pipeline component • Link together ML components
LESSON TWO	**Data Exploration and Preparation**	• Execute and track the Exploratory Data Analysis (EDA) • Clean and pre-process the data • Segregate (split) datasets
LESSON THREE	**Data Validation**	• Use pytest with parameters for reproducible and automatic data tests • Perform deterministic and non-deterministic data tests
LESSON FOUR	**Training, Validation and Experiment Tracking**	• Tame the chaos with experiment, code and data tracking • Track experiments with W&B • Validate and choose best-performing model • Export model as an inference artifact • Test final inference artifact
LESSON FIVE	**Release and Deploy**	• Release pipeline code • Options for deployment and how to deploy a model

Course 3: Deploying a Scalable ML Pipeline in Production

This course teaches students how to deploy a machine learning model into production. En route to that goal, students will learn how to put the finishing touches on a model by taking a fine-grained approach to model performance, checking bias and ultimately writing a model card. Students will also learn how to version control their data and models using Data Version Control (DVC). In the last piece of preparation for deployment, students will learn Continuous Integration and Continuous Deployment accomplished using GitHub Actions and Heroku. Finally, students will learn how to write a fast, type-checked and auto-documented API using FastAPI.

Course Project : Deploying a Machine Learning Model on Heroku with FastAPI

In this project, students will deploy a machine learning model on Heroku. The students will use Git and DVC to track their code, data and model while developing a simple classification model on the Census Income Data Set. After creating the model, the students will finalize the model for production by checking its performance on slices and writing a model card encapsulating key knowledge about the model. Students will put together a Continuous Integration and Continuous Deployment framework and ensure their pipeline passes a series of unit tests before deployment. Lastly, an API will be written using FastAPI and tested locally. After successful deployment, the API will be tested live using the requests module.

After completion, you will have a working API that is live in production, a set of tests, a model card and a full CI/CD framework. On its own, this project can be a portfolio piece but can also be applied to other projects, e.g., continuous integration, to flesh them further out.

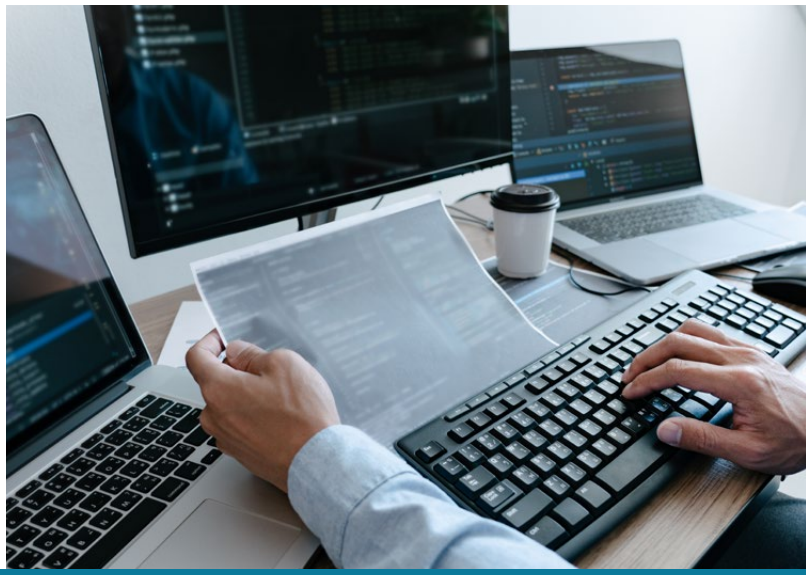
		LEARNING OUTCOMES
LESSON ONE	**Performance Testing and Preparing a Model for Production**	• Analyze slices of data when training and testing models • Probe a model for bias using common frameworks such as Aequitas • Write model cards that explain the purpose, provenance and pitfalls of a model
LESSON TWO	**Data and Model Versioning**	• Version control data/models/etc locally using DVC • Set up remote storage for use with DVC • Create pipelines and track experiments with DVC
LESSON THREE	**CI/CD**	• Follow software engineering principles by automating, testing and versioning code • Set up Continuous Integration using GitHub Actions • Set up Continuous Deployment using Heroku
LESSON FOUR	**API Deployment with FastAPI**	• Write an API for machine learning inference using FastAPI • Deploy a machine learning inference API to Heroku • Write unit tests for APIs using the requests module

Course 4: Automated model scoring and monitoring

This course will help students automate the DevOps processes required to score and re-deploy ML models. After model deployment, you will set up regular scoring processes, learn to reason carefully about model drift, and whether models need to be retrained and re-deployed. Students will learn to diagnose operational issues with models, including data integrity and stability problems, timing problems and dependency issues. Finally, students will learn to set up automated reporting with APIs.

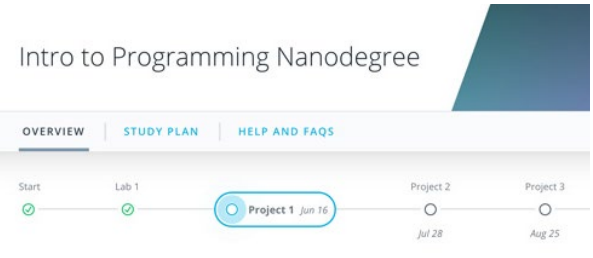
Course Project :
A Dynamic Risk Assessment System

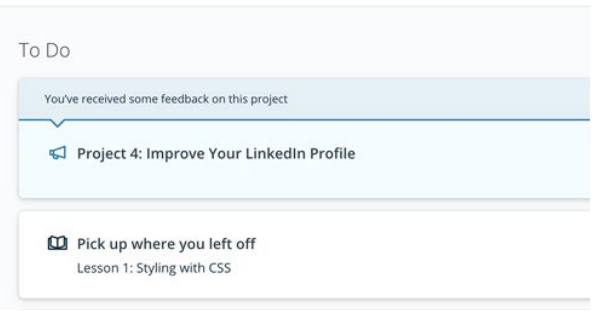
In this project, you will make predictions about attrition risk in a fabricated dataset. Begin by setting up processes to ingest data and score, retrain and re-deploy ML models that predict attrition risk while writing scripts that automatically check for new data and model drift. You'll also set up APIs that allow users to access model results, metrics and diagnostics. After completing this project, students will have an end-to-end, automated ML project that performs risk assessments. This project can be a valuable addition to students' portfolios, and the concepts they apply in the project can be applied to business problems across a variety of industries.

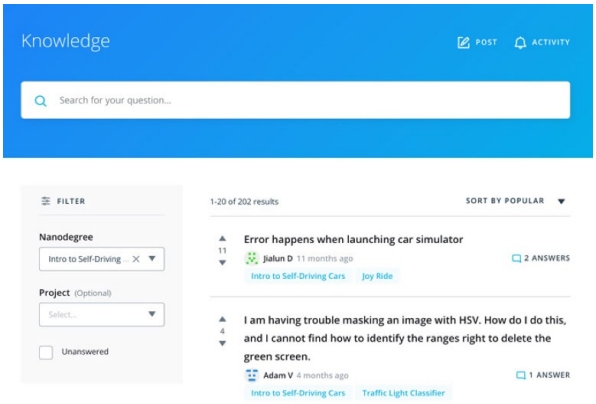


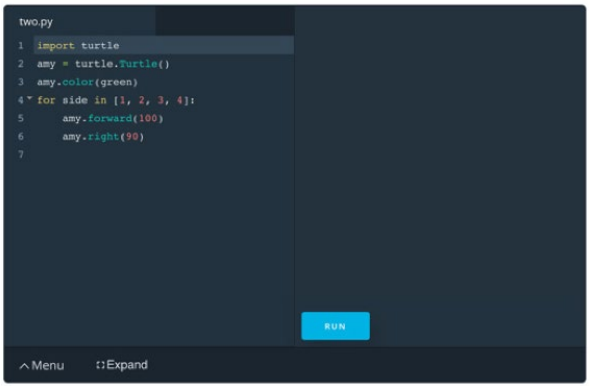
		LEARNING OUTCOMES
LESSON ONE	**Model Training and Deployment**	• Ingest data • Automatically train models • Deploy models to production • Keep records about processes • Automate processes using cron jobs
LESSON TWO	**Model Scoring and Model Drift**	• Automatically score ML models • Keep records of model scores • Check for model drift using several different model drift tests • Determine whether models need to be retrained and re-deployed
LESSON THREE	**Diagnosing and Fixing Operational Problems**	• Check data integrity and stability • Check for dependency issues • Check for timing issues • Resolve operational issues
LESSON FOUR	**Model Reporting and Monitoring with APIs**	• Create API endpoints that enable users to access model results, metrics and diagnostics • Set up APIs with multiple, complex endpoints • Call APIs and work with their results

Our Classroom Experience









REAL-WORLD PROJECTS

Build your skills through industry-relevant projects. Get personalized feedback from our network of 900+ project reviewers. Our simple interface makes it easy to submit your projects as often as you need and receive unlimited feedback on your work.

KNOWLEDGE

Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students and discover in real-time how to solve the challenges that you encounter.

STUDENT HUB

Leverage the power of community through a simple, yet powerful chat interface built within the classroom. Use Student Hub to connect with your technical mentor and fellow students in your Nanodegree program.

WORKSPACES

See your code in action. Check the output and quality of your code by running them on workspaces that are a part of our classroom.

QUIZZES

Check your understanding of concepts learned in the program by answering simple and auto-graded quizzes. Easily go back to the lessons to brush up on concepts anytime you get an answer wrong.

CUSTOM STUDY PLANS

Work with a mentor to create a custom study plan to suit your personal needs. Use this plan to keep track of your progress toward your goal.

PROGRESS TRACKER

Stay on track to complete your Nanodegree program with useful milestone reminders.

Learn with the Best

Joshua Bernhard

DATA SCIENTIST AT THUMBTACK

Josh has been sharing his passion for data for nearly a decade at all levels of university, and as a Data Science Instructor for coding bootcamps. He's used data science for work ranging from cancer research to process automation.

Giacomo Vianello

LEAD DATA SCIENTIST

Giacomo is an end-to-end data scientist with a passion for state-of-the-art but practical technical solutions. He is Lead Data Scientist at Cape Analytics, where he develops AI systems to extract intelligence from geospatial imagery bringing cutting-edge AI solutions to the insurance and real estate industries.

Justin Clifford Smith, Ph.D.

SENIOR DATA SCIENTIST

Justin is a Senior Data Scientist at Optum where he works to make healthcare more efficient with natural language processing and machine learning. Previously he was a Data Scientist at the US Census Bureau. His doctorate is from the University of California, Irvine where he studied theoretical physics.

Bradford Tuckfield

DATA SCIENTIST AND WRITER

Bradford is a data scientist and writer. He has worked on applications of data science in a variety of industries. He's the author of Dive Into Algorithms, forthcoming with No Starch Press.

Learn with the Best

Ulrika Jägare

HEAD OF AI/ML STRATEGY EXECUTION IN ERICSSON

Ulrika has been with Ericsson for 21 years in various leadership roles, out of which 11 years in the Data and AI space. Ulrika holds a Master of Science degree from University of Lund in Sweden and is also author of seven published books in Data Science.

All Our Nanodegree Programs Include:

EXPERIENCED PROJECT REVIEWERS

REVIEWER SERVICES

- Personalized feedback & line by line code reviews
- 1600+ Reviewers with a 4.85/5 average rating
- 3 hour average project review turnaround time
- Unlimited submissions and feedback loops
- Practical tips and industry best practices
- Additional suggested resources to improve

TECHNICAL MENTOR SUPPORT

MENTORSHIP SERVICES

- Questions answered quickly by our team of technical mentors
- 1000+ Mentors with a 4.7/5 average rating
- Support for all your technical questions

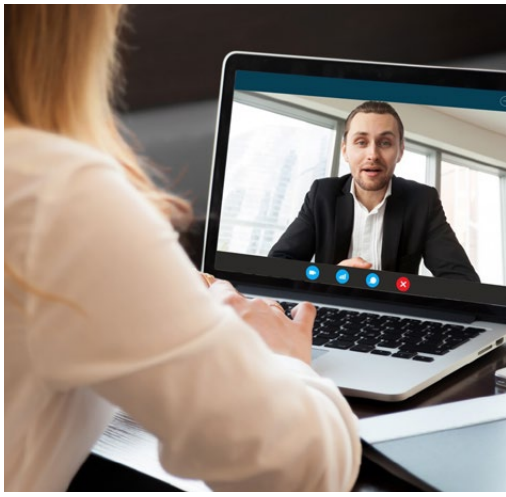
PERSONAL Ccareer SERVICES

CAREER SUPPORT

- Resume support
- Github portfolio review
- LinkedIn profile optimization

Frequently Asked Questions

PROGRAM OVERVIEW



WHY SHOULD I ENROLL?

Data and AI professionals today are expected to be able to go beyond training ML models to packaging, deploying, and monitoring them in production environments. Whether you're a Data Scientist, Data Engineer, Software Engineer, or any other role working with ML models, building this DevOps skillset will set you apart.

WHAT JOBS WILL THIS PROGRAM PREPARE ME FOR?

The skills you build in this program will be instrumental in roles such as Data Scientist, Data Engineer, Machine Learning Engineer, DevOps Engineer, and beyond.

ML DevOps is leveraged in a wide range of industries, from public transportation and healthcare to engineering, safety, and manufacturing. From models that automatically recognize certain types of medication to models that anticipate the effects of earthquakes, autonomous (and deployed!) systems yield real-world impact with the assistance of MLOps.

HOW DO I KNOW IF THIS PROGRAM IS RIGHT FOR ME?

This course is for individuals who recognize the importance of machine learning model deployment but struggle to push the models they have developed in modeling environments to production to be self-functioning.

ENROLLMENT AND ADMISSION

DO I NEED TO APPLY? WHAT ARE THE ADMISSION CRITERIA?

No. This Nanodegree program accepts all applicants regardless of experience and specific background.

WHAT ARE THE PREREQUISITES FOR ENROLLMENT?

A well-prepared student will already be familiar with:

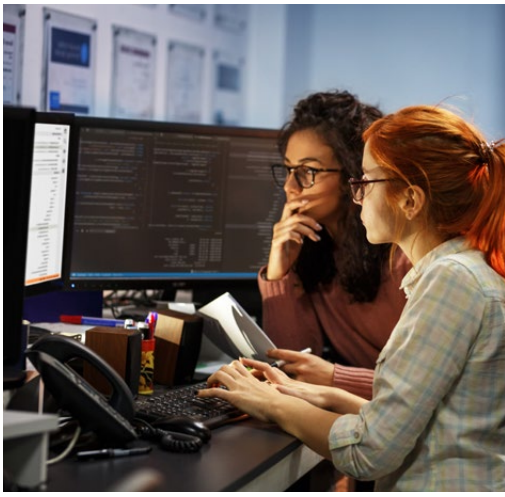
- The data science process and overall workflow of building machine learning models
- Using Jupyter notebooks to solve data science-related problems
- Writing scripts using NumPy, pandas, Scikit-learn, TensorFlow/PyTorch in Jupyter notebooks that clean data (as part of ETL), feed it into a machine learning model and validate the performance of the model
- Using the Terminal, version control in Git, and using GitHub

IF I DO NOT MEET THE REQUIREMENTS TO ENROLL, WHAT SHOULD I DO?

To prepare, we recommend the **Introduction to Machine Learning** and **AI Programming with Python** programs, to build your comfortability with ML concepts and using python in an AI context.

FAQs Continued

TUITION AND TERM OF PROGRAM



HOW IS THIS NANODEGREE PROGRAM STRUCTURED?

The Machine Learning DevOps Engineer Nanodegree program is comprised of content and curriculum to support four (4) projects. We estimate that students can complete the program in four (4) months working 10 hours per week.

Each project will be reviewed by the Udacity reviewer network. Feedback will be provided and if you do not pass the project, you will be asked to resubmit the project until it passes.

HOW LONG IS THIS NANODEGREE PROGRAM?

Access to this Nanodegree program runs for the length of time specified above. If you do not graduate within that time period, you will continue learning with month to month payments. See the **Terms of Use** and **FAQs** for other policies regarding the terms of access to our Nanodegree programs.

CAN I SWITCH MY START DATE? CAN I GET A REFUND?

Please see the Udacity Nanodegree program **FAQs** for policies on enrollment in our programs.

SOFTWARE AND HARDWARE

WHAT SOFTWARE AND VERSIONS WILL I NEED IN THIS PROGRAM?

You will need a computer running a 64-bit operating system with at least 8GB of RAM, along with administrator account permissions sufficient to install programs including Anaconda with Python 3.x and supporting packages.

Most modern Windows, OS X, and Linux laptops or desktops will work well; we do not recommend a tablet since they typically have less computing power. We will provide you with instructions on how to install the required software packages. Additional tech requirements can be found at **https://www.udacity.com/tech/requirements**.