# ML Engineer Program Syllabus

## About FourthBrain

FourthBrain educates aspiring Machine Learning engineers on the technical and practical skills required to provide immediate value to an AI product development or R&D team. Our programs emphasize open collaboration and communication, and our unique approach to cohort-based learning is aimed at helping you achieve your personal career goals while giving you the best chance to develop strong, lasting relationships with other ML practitioners on similar journeys.

## Your Learning Transformation

By the end of this course, you will be able to contribute to high-performing AI product teams by leveraging real-world data to **build**, **package**, and **deploy** state-of-the-art ML models as containerized web applications in cloud-based production environments.

## Capstone Project

Capstone projects are designed to demonstrate your understanding of MLE software development and its implications. This includes understanding the potential business-value of your application and its extensibility, as well as the tools required to build, optimize, package, and deploy your ML models in a production software development environment. Projects are typically developed in groups of two or three; solo projects are possible, especially in situations where external support (i.e., your colleagues or other collaborators are available). Final deliverables will include a final presentation and GitHub repo, which together will allow you to share the details of what you've accomplished with your potential users, collaborators, employers, or the wider open-source ML community. You can see past capstone projects here.

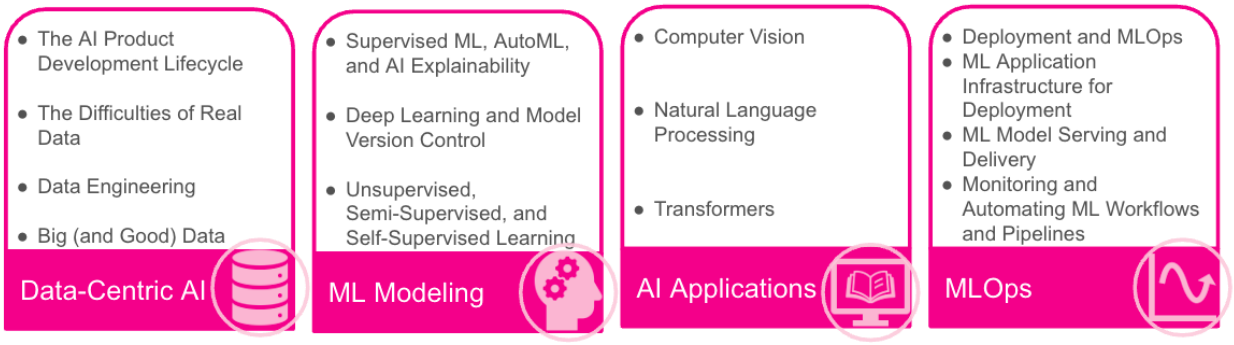
## Career Growth

The Machine Learning Engineer program is designed for you to acquire the skills and knowledge required to work on a Machine Learning Engineering team. Whatever your reason for taking the program - to get a new role at a new company, to gain skills for your current job, or just for fun - we will support your career growth by helping you connect to professionals and employers, via guest speaking events and inviting employers to the final project presentation day. Career services assistance is available during the program and after graduation to help ensure that all candidates achieve their career goals.

## Program Format

Each week, we will curate content related to the weekly topic for you to review asynchronously. You will come together with your instructors and peers to discuss, contextualize, and practice the concepts for the week.

The Machine Learning Engineer program is divided into four pillars:  Data Centric AI, Machine Learning Modeling, AI Applications, and MLOps.




During our first week we kick off the course with a review of important ML software engineering tools.  We get our development environment set up and test out that we can build basic ML models.

Our last week is completely focused on Demo Day for your projects!

# Weekly Schedule

| Week | Live Session Concepts | Coding Exercise(s) | Capstone |
|---|---|---|---|
| Week 1<br><br>Getting Started with ML Engineering | • Program Overview<br>• Software Development Environment setup | MLE Basic *Tools*:<br>• Unix CLI, Git, Conda, Pip, Jupyter<br>• Pandas, Matplotlib, Seaborn, Sci-Kit Learn | • Networking |
| Week 2<br><br>AI Product Development | • AI Product Development Lifecycle & ML Project Scoping<br>• Data Centric AI<br>• Responsible ML Principles | • Creating a sentiment analyzer<br>• *Tools*: Hugging Face Transformers Library, Twitter API | • Ideation Workshop<br>• AI Product/ML Project Scoping |
| Week 3<br><br>The Difficulties of Real Data | • Best Practices for High-Quality Data<br>• REST APIs<br>• Fine-Tuning Pre-Trained Models<br>• Building ML POCs and MVPs | • Collect real data from Twitter and Reddit APIs<br>• Fine-Tune a pre-trained transformer for sentiment analysis<br>• Develop a data-centric proof of concept<br>• *Tools*: Hugging Face Trainer API, Reddit API | • Peer Review and Teaming |

| | | | |
|---|---|---|---|
| Week 4<br><br>Data Engineering | • Data Engineering Workflows<br>• Data Wrangling & Exploratory Data Analysis<br>• Feature Selection & Engineering<br>• Data Leakage<br>• Building ETL Pipelines | • Exploring and wrangling structured data to predict sales with simple ML pipeline<br>• Track and manage datasets<br>• Build an ETL workflow<br>• *Tools*: Airflow, DVC | • Project Proposals<br>• Project Pitch Day I |
| Week 5<br><br>Big (and Good) Data | • Types of Distributed Computing for ML<br>• Data Preparation<br>• Big Data Tool Landscape<br>• How Good Data Becomes Big Data | • Build an ML pipeline and perform distributed hyperparameter tuning to predict subscriptions & promotions<br>• *Tools*: Spark, Delta Lake, MLlib | • Exploratory Data Analysis and Data Lineage Documentation |
| Week 6<br><br>Supervised ML, AutoML, and Explainability | • Essential Regression & Classification Algorithms<br>• Data Imbalance<br>• Accuracy Metrics<br>• AutoML Libraries<br>• AI Explainability | • Detect data imbalances and predict electronics purchases with explainable ML pipelines<br>• Perform AutoML to search for an optimal tree-based pipeline<br>• *Tools*: TPOT, SHAP, Streamlit | • Establish ML Modeling Baseline |

| | | | |
|---|---|---|---|
| Week 7<br><br>Deep Learning and Model Version Control | • Neural Network Basics<br>• Recurrent Neural Networks (RNNs)<br>• Long Short-Term Memory Networks (LSTMs)<br>• Generative Adversarial Networks (GANs)<br>• ML Model Version Control | • Build, train, and evaluate a neural network for fuel efficiency prediction from scratch<br>• Compare neural network performance based on type<br>• *Tools*: MLflow Tracking | • Initial GitHub Documentation |
| Week 8<br><br>Unsupervised, Semi-Supervised, and Self-Supervised Learning | • Dealing with Unstructured Data<br>• Clustering<br>• Dimensionality Reduction<br>• Label propagation/label spreading<br>• Co-training algorithms<br>• Zero-shot learning | • Performing Customer Segmentation<br>• Predicting product sales using semi-supervised learning<br>• *Tools*: Sk-learn (pca, Kmeans, Silhouette Analysis) | • Data & Model Iterations |
| Week 9<br><br>Computer Vision | • Convolutional Neural Networks (CNNs)<br>• Computer Vision Benchmarks<br>• Dealing with Images<br>• Object Detectors<br>• Semantic Segmentation<br>• Explainability & Saliency | • Few-shot dog detection dogs through fine-tuning of a pre-trained single shot multibox detector<br>• *Tools*: TensorFlow Model Garden | |

| | | | |
|---|---|---|---|
| Week 10<br><br>Natural Language Processing | • Natural Language Benchmarks<br>• Dealing with Text<br>• Tokenization & Word Embeddings<br>• Bag of Words, Term Frequency Inverse Document Frequency<br>• Using Pre-Trained Word Embeddings<br>• Named Entity Recognition | • Analyze tweets using basic NLP tasks<br>• Build hate speech detectors using a naive Bayes classifier and a bidirectional LSTM<br>• *Tools:* Tensorboard Embedding Projector, LIME Text Explainer | • Project Pitch Day II |
| Week 11<br><br>Transformers | • Encoder and Decoder Networks<br>• Bidirectional Encoder Representations from Transformers (BERT)<br>• General Pre-Trained Transformers (GPT-3)<br>• Fine-Tuning of Pre-Trained Transformers | • Build a transformer model for news article text classification from scratch<br>• Fine-tune a pre trained transformer<br>• *Tools:* Hugging Face Models | • Project Deployment Demonstration |
| Week 12<br><br>Deployment and MLOps | • Web Application Frameworks<br>• ML Inference (Batch, Streaming, Real-Time, Edge)<br>• MLOps Level 0: Manual | • Deploy a public API stock prediction service<br>• *Tools*: VS Code, FastAPI, AWS | • GitHub Repository Revisions |

| | | | |
|---|---|---|---|
| Week 13<br><br>ML Application Infrastructure for Deployment | • Operating Systems and Virtual Machines<br>• Containers and Container Orchestration<br>• Model Management and Model Registries | • Build and deploy a containerized stock prediction service<br>• Manage model versions with a model registry<br>• *Tools*: Docker, MLflow Model Registry | • 1-Page Narrative and Infrastructure Diagramming |
| Week 14<br><br>ML Model Serving and Delivery | • Model Servers, Architecture, and Platforms<br>• Continuous Integration and Delivery (CI/CD) | • Implement stock prediction app within a serverless computing service<br>• Deploy stock prediction service<br>• *Tools*: AWS Lambda, API Gateway | • Final Feature Additions |
| Week 15<br><br>Monitoring and Automating ML Workflows and Pipelines | • ML Monitoring & Observability<br>• MLOps L1: Pipeline Automation<br>• Full-Stack MLOps in the Cloud | • Build, train, optimize, deploy, and monitor a salary prediction app on AWS<br>• *Tools*: AWS SageMaker, CloudWatch | • Final Capstone Presentation Submission |
| Week 16<br><br>Demo Week! | • Storytelling Workshop<br>• Capstone Presentation Practice | • Final Capstone GitHub submission! | • Demo Day! |

Apply today to the Machine Learning Engineer Program!