Machine Learning School

Welcome to the Machine Learning School!

When I started thinking about this program, I wanted to create a cohort focusing on teaching how to build production-ready Machine Learning systems, but I ended up building a community of like-minded people who'll be pushing the industry forward for the next 10-20 years.

I'm so lucky!

The Machine Learning School is much more than it was meant to be, and that's thanks to you and every person involved in this journey.

This guide will help you get ready and navigate the program.

Strap on and have fun!

The Community

We use Circle to run the community. That's the place where everything happens!

These are the main sections on the community website and what you'll find on each of them:

- **Schedule**. Here you'll find every upcoming and past event in the community. Every session of the cohort will be scheduled here. We often run one-off sessions about different topics, and they will also appear here.
- **Discussions**. This is a discussion forum where the main conversation happens. This is where you'll post questions, help others, and learn the most.
- **Cohort N**. There will be a section dedicated to every cohort. Past cohorts will be under the Archive section.

Here are a few recommendations on how to use the community:

- Download the Circle mobile app. You can download the iOS or Android version. I find the web application much more comfortable, but the mobile app is usually more convenient.
- Update your member profile. Add a professional-looking photo and as much information as you are comfortable sharing. The community is a great place to find job opportunities, and if you are interested, you don't want to remain anonymous.
- Whenever you have a question, ask it publicly in the Discussions forum. This increases the chances you'll get an answer, and it will help other people with the same question.
- Asking questions will force you to think deeply about the material. I recommend you attend sessions with pen and paper and write down anything unclear. You can write a public post with your notes at the end of each session. I find this helps tremendously.
- Try to help your classmates by answering their questions whenever you can. Everyone is at a different stage in their careers, and supporting each other is how we progress most.
- If you need help with your account or want to talk privately, send me a direct message. I'll get back to you as soon as I get it.

Before The Cohort Starts

Here is a checklist of everything you should do before your cohort starts. These will help you get ready and get the most out of the program:

1. Say hello and introduce yourself to the community. It's a nice, personal touch. Use this opportunity to let everyone know what you do and what you are looking forward to achieving.
2. RSVP to the cohort sessions you are planning to attend. You can do this under the Schedule section on the community site. This will ensure you get notifications related to the cohort and help me understand how many people will join every session.

3. Fork the cohort's GitHub Repository containing the source code of the program. This will create your repository to make changes and try new things.
4. Create a brand new AWS account. This is usually more straightforward than using an existing account, especially if your employer owns it. You'll need to set up permissions and services, and it's easier when you don't need to ask for authorization.
5. Watch the "Introduction to Amazon SageMaker Studio" video to get a general understanding of the platform.
6. Create an Amazon SageMaker domain. The "Getting Started on Amazon SageMaker Studio" video will walk you through the process. Clone the repository you forked from inside SageMaker Studio.
7. Run the "Initial Setup" section of the cohort notebook and set up your Execution Role with the appropriate permissions indicated in the notebook.

Cohort Sessions

The program consists of six live sessions of 90 minutes each. We'll run the sessions using Google Meet, and you can attend live or watch the recording later.

You'll find the Google Meet URL of every session inside the community's Schedule section.

The recordings of every session will be posted a few hours after the session finishes. You'll find every recording for the running cohort inside the community's Recordings section.

It's All About Penguins

We'll work on a simple problem throughout the program: a dataset with different characteristics of three species of penguins. We'll set up a multi-class classification model that uses the measurements of a penguin and predict its species.

If you are interested, this GitHub Repository contains more information about the dataset we'll use.

This program focuses on setting up a production-ready Machine Learning pipeline, so I wanted to keep the underlying problem and the model as simple as possible.

We'll talk about some fundamental principles of Machine Learning, but generally, we'll keep those conversations light and focus primarily on engineering techniques to build production pipelines.

Source Code

This is the GitHub Repository containing the source code of the program. It's a public repository; fork it to follow along.

Every cohort is a new opportunity to improve the material and code, so don't be surprised if the code changes frequently between sessions of the cohort. It's a good idea to update your copy before every session to ensure you have the latest version.

If you find any problems with the code or have any ideas on improving it, please, share your recommendations in the Discussions forum.

Assignments

You'll find a list of assignments under every session in the program. These assignments take you off your comfort zone and help you practice the main ideas we discussed during each session.

You don't have to solve every assignment. Instead, choose the ones that you think will help you the most. For example, some assignments will ask to replace the TensorFlow model we built with a PyTorch equivalent. You don't have to solve these assignments if you aren't interested in PyTorch.

Before working on the assignments, ensure you can run the session code successfully.

Feel free to use the Discussions forum if you get stuck solving any assignments.

The first session introduces the problem we want to solve and builds a simple pipeline with one step to preprocess the dataset.

Session 1 - Building a Pipeline

This session will introduce the program and start building the production pipeline. We'll cover the following topics:

1. Introduction to the program.
2. An application about Penguins.

3. Introduction to Machine Learning Pipelines.
4. Designing a production pipeline.
5. SageMaker Processing Jobs and the Processing Step.
6. Transforming and splitting the Penguins dataset.
7. Configuration and caching of pipelines.

Assignments

1. Throughout the course, you will work on the "Pipeline of Digits" project to set up a SageMaker pipeline for a simple computer vision project. For this assignment, open the `mnist.ipynb` notebook and follow the instructions to prepare everything you need to start the project.
2. Set up a SageMaker pipeline for the "Pipeline of Digits" project. Create a Processing Step where you split 20% off the MNIST train set to use as a validation set.

Session 2 - Training and Tuning

This session will extend the pipeline with a step for training a model. We'll cover the following topics:

1. Training and tuning in production systems.
2. SageMaker Training Jobs and the Training Step.
3. SageMaker Hyperparameter Tuning Jobs and the Tuning Step.
4. A multi-class classification network to predict species of penguins.
5. Implicit and explicit dependencies between pipeline steps.

Assignments

1. Modify the training script to accept the `learning_rate` as a new hyperparameter.
2. If you prefer PyTorch, replace the TensorFlow Estimator with a PyTorch Estimator. Check the Use PyTorch with the SageMaker Python SDK page for an example of creating a PyTorch Estimator.

3. Modify the Hyperparameter Tuning Job to find the best `learning_rate` value between `0.01` and `0.03`. Check the ContinuousParameter class for more information on how to configure this parameter.

4. Modify the SageMaker Pipeline you created for the "Pipeline of Digits" project and add a Training Step. This Training Step should receive the train and validation splits.

Session 3 - Evaluation and Registration

This session will extend the pipeline with a step for evaluating the model and another for registering it in the Model Registry. We'll cover the following topics:

1. Model versioning in production systems.
2. Evaluating the Penguins model.
3. Introduction to the Model Registry.
4. The SageMaker Model Step.
5. The SageMaker Condition Step and Fail Step.

Assignments

1. The evaluation script produces an evaluation report containing the accuracy of the model. Extend the evaluation report by adding other metrics. For example, add the support of the test set (the number of samples in the test set.)

2. Modify your pipeline to add a new Condition Step that's called if the model's accuracy is not above the specified threshold. Set the condition to succeed if the accuracy is above 50% and register the model as "PendingManualApproval." Don't register the model if the accuracy is not greater or equal to 50%. In summary, register the model as "Approved" if its accuracy is greater or equal to 70% and as "PendingManualApproval" if its accuracy is greater or equal to 50%.

3. If you run the Training and Tuning Steps simultaneously, create two different Evaluation Steps to evaluate both models independently.

4. Instead of running the Training and Tuning Steps simultaneously, run the Tuning Step but create two evaluation steps to evaluate the two best models produced by

the Tuning Step. Check the TuningStep.get_top_model_s3_uri() function to retrieve the two best models.

5. Modify the SageMaker Pipeline you created for the "Pipeline of Digits" project and add an evaluation and a registration step.

Session 4 - Deploying the Model

This session will extend the pipeline with a step for deploying the model to an endpoint. We'll cover the following topics:

1. Deploying directly from the Model Registry.
2. Custom inference code.
3. Introduction to model repacking in SageMaker.
4. Automatically capturing live traffic.
5. The SageMaker Lambda Step.
6. Extending the Pipeline to deploy the model.

Assignments

1. Our custom inference code doesn't support processing more than one sample simultaneously. Modify the inference script to allow the processing of multiple samples simultaneously. The output should be an array of JSON objects containing the prediction and the confidence corresponding to each input sample.
2. Load the test data and run every sample through the endpoint using a Predictor. Build a simple function that computes the accuracy of this test set.
3. Customize the inference process of the "Pipeline of Digits" project endpoint to receive a JSON containing an image URL and return the digit in the image.
4. Modify the SageMaker Pipeline you created for the "Pipeline of Digits" project and add a Lambda Step to deploy the model automatically.

Session 5 - Data Monitoring

This session extends the pipeline to computing a data baseline and sets up a Data Monitoring Job to detect anomalies with live traffic data.

1. Identifying data drift from first principles.
2. Computing a data baseline to detect data drift.
3. The SageMaker QualityCheck Step.
4. Setting up a Data Monitoring Schedule.

Assignments

1. Modify the SageMaker Pipeline you created for the "Pipeline of Digits" project and add the necessary steps to generate a Data Quality baseline.
2. Build a simple function that generates fake traffic to the "Pipeline of Digits" endpoint so we can start monitoring the quality of the data coming in.
3. Modify the SageMaker Pipeline you created for the "Pipeline of Digits" project and automatically add a new Lambda Step to schedule the Data Quality Monitoring Job.

Session 6 - Model Monitoring

This session extends the pipeline to computing a performance baseline and sets up a Model Monitoring Job to detect any drift or anomalies with the model predictions.

We will cover the following topics:

1. Identifying model drift from first principles.
2. Computing a performance baseline to detect model drift.
3. SageMaker Batch Transform Jobs and the Transform Step.
4. Generating ground-truth data.
5. Computing performance metrics.
6. Setting up a Model Monitoring Schedule.

Assignments

1. Modify the SageMaker Pipeline you created for the "Pipeline of Digits" project and add the necessary steps to generate a Model Quality baseline.
2. Build a simple function that generates fake ground truth data for the data captured by the "Pipeline of Digits" endpoint.

3. Modify the SageMaker Pipeline you created for the "Pipeline of Digits" project and automatically add a new Lambda Step to schedule the Model Quality Monitoring Job.

Additional Resources

- Designing Machine Learning Systems: An Iterative Process for Production-Ready Applications by Chip Huyen.