MLOps Bootcamp Syllabus

Machine Learning Architects Basel

Bassem Ben Hamed

December 2022

ML ARCHITECTS
BASEL

Agenda

- **Training overview**
- **Syllabus**

ML ARCHITECTS BASEL

Training overview

- **Objective :**

This training aims to teach practical aspects of productionizing ML services — from collecting requirements to model deployment and monitoring.

- **Target audience:**

Data scientists and ML engineers. Also, software and data engineers interested in learning about putting ML in production.

- **Pre-requisites:**

- Python
- Docker
- Being comfortable with command line
- Knowledge on machine learning
- Prior programming experience (at least 1+ year)

Syllabus :

Module 1 : Data, Model & Experiment Management :

- **Introduction and data understanding**

- What does a typical ML pipeline look like ?
- What is MLOps?
- Why do we need MLOps ?
- MLOps maturity levels
- Course overview
- Environment preparation (install Git , Docker ..)
- Understand the data and the use case (Bank transaction fraud detection)
- Exploratory data analysis and extract insight from the data

- **Data cleaning and modeling**

- Understand the techniques of data cleaning and preprocessing
- Understand how to deal with unbalanced data
- Understand how the specified ML models works (Random forest, Logistic regression & XGboost)
- Train and evaluate ML models

- **Data, code, experiment versioning**

- Setup our version control system (DAGsHub)
- Why we need to version data ?
- Version data using DVC
- Why we need experiment tracking?
- Configure mlflow experiment tracking tool with the training notebook

Syllabus :

Module 2 : Tooling, Infrastructure & Deployment :

- **Backend and Frontend parts**

From jupyter notebooks to modular code
- Backend part
→ Why Fast API as a web framework?
→ Create restAPI using FastApi
- Frontend part
→ Create user interface using streamlit

- **Deploy the application**

- Docker for packaging the application
- Build the docker images for the backend and frontend
- Orchestrate between the backend and front-end locally using docker compose
- How to deploy the application
- What is AWS?
- send the docker images to AWS ECR (Elastic container registry)
- deploy the backend on AWS ECS

- **CI/CD/CT pipeline**

- deploy the frontend on AWS ECS
- Ensure the connection between backend and frontend
- Continuous delivery CI/CD/CT pipelines
- what is Github actions
- Configure CI/CT/CD pipelines

Syllabus :

Module 3 : Serving and Testing and Validating the ML system

- **Testing**

-Testing: unit, integration
-Testing data and models using Deepchecks

- **Monitoring**

- Monitoring ML-based services
- Configure Arize AI as an ML monitoring system