MLOps & Data Science Professional Program

MLOps & Data Science Professional Program

3-Month Comprehensive Curriculum

Month 1: Data Science Foundations & ML Model Building

Goal: Master the end-to-end data science lifecycle from data preparation to model training and evaluation.

Week 1: Applied Statistics & Python for DS

- **Python for Data Science:** Setting up environments and utilizing core libraries for data manipulation and scientific computing.
- **Statistical Analysis:** Applying fundamental statistical concepts (mean, variance, distributions) using the `SciPy` library.
- **Data Wrangling:** Mastering `NumPy` for high-performance array operations and `Pandas` for cleaning, filtering, and aggregating structured datasets.

Week 2: Machine Learning Fundamentals

- **Core ML Concepts:** Understanding the theoretical differences and use cases for Supervised vs. Unsupervised learning.
- **Supervised Learning:** Building, training, and interpreting regression and classification models (e.g., Linear Regression, Random Forests, SVM) using `Scikit-Learn`.
- **Unsupervised Learning:** Implementing clustering algorithms (like K-Means) and dimensionality reduction techniques (like PCA) for pattern discovery.

Week 3: Deep Learning Basics

- **Neural Network Architecture:** Introduction to perceptrons, hidden layers, activation functions, and the mathematics of Deep Learning.
- **Optimization & Training:** Understanding gradient descent, backpropagation, and loss functions to optimize model weights.
- **Framework Mastery:** Building and training foundational neural networks utilizing industry-standard frameworks like `TensorFlow` or `PyTorch`.

Week 4: Model Evaluation & Tuning

- **Performance Metrics:** Analyzing and interpreting critical ML metrics including Accuracy, Precision, Recall, F1-score, and ROC-AUC curves.
- **Validation Strategies:** Implementing K-fold Cross-Validation to ensure model generalization and prevent overfitting.
- **Hyperparameter Tuning:** Utilizing Grid Search and Random Search techniques to systematically find the optimal parameters for model performance.

Month 2: ML Model Deployment & Containerization

Goal: Transition models from local Jupyter Notebooks to scalable, production-ready, containerized web services.

Week 5: Model Serialization & API Development

- **Model Exporting:** Saving trained machine learning models efficiently using serialization libraries like `Pickle` or `Joblib`.
- **REST API Fundamentals:** Understanding the principles of RESTful architecture, HTTP methods, and request/response lifecycles.
- **Serving Predictions:** Building lightweight, high-performance APIs using `FastAPI` or `Flask` to expose ML models to web or mobile frontends.

Week 6: Containerization for ML

- **Docker Basics:** Introduction to containerization concepts and understanding the differences between Docker and traditional virtual machines.
- **Writing Dockerfiles:** Crafting optimized `Dockerfiles` specifically for ML environments, ensuring all dependencies and libraries are accurately resolved.
- **Multi-Container Apps:** Utilizing `Docker Compose` to orchestrate multi-container setups (e.g., connecting a FastAPI application with a Redis cache).

Week 7: Version Control for Data & Models

- **Data Versioning:** Implementing Data Version Control (DVC) to track changes in large datasets and machine learning models alongside code.
- **Experiment Tracking:** Setting up and utilizing `MLflow` to log parameters, code versions, metrics, and output files for every training run.
- **Model Registry:** Managing the lifecycle of ML models, transitioning them systematically from staging to production environments.

Week 8: Cloud ML Platforms

- **Cloud ML Concepts:** Overview of managed machine learning services and the benefits of cloud infrastructure for scalable training.
- **AWS SageMaker:** Introduction to Amazon SageMaker for provisioning notebook instances, distributed training, and model hosting.
- **Endpoint Creation:** Deploying trained models as scalable, highly available endpoints using AWS SageMaker or Google Cloud Vertex AI.

[Project 1: Containerized Computer Vision API]

- **Scope:** Train a custom image classification model using PyTorch, serialize the model, containerize the entire application using Docker, and serve real-time predictions via a high-performance FastAPI endpoint.

Month 3: CI/CD for Machine Learning & Operations

Goal: Automate the entire ML lifecycle, monitor production models for degradation, and ensure continuous delivery (CD) and continuous training (CT).

Week 9: CI/CD Pipelines for ML

- **Automation Basics:** Understanding Continuous Integration and Continuous Deployment concepts specifically tailored for Machine Learning workflows.
- **GitHub Actions:** Writing YAML workflows to automate code linting, unit testing, and model evaluation upon every code commit.
- **Automated Delivery:** Configuring pipelines to automatically build Docker images and push them to container registries (like Docker Hub or AWS ECR) post-testing.

Week 10: Model Monitoring & Drift Detection

- **Understanding Drift:** Identifying and differentiating between Concept Drift (changes in the target variable) and Data Drift (changes in input data distributions).
- **Metrics Collection:** Instrumenting ML APIs to expose critical performance and operational metrics using `Prometheus`.
- **Live Dashboards:** Connecting Prometheus to `Grafana` to build real-time, interactive dashboards for monitoring model health, latency, and prediction distributions.

Week 11: Orchestrating ML Workflows

- **Workflow Orchestration:** The need for complex, multi-step pipeline management in enterprise MLOps.
- **Advanced Tooling:** Introduction to advanced orchestration frameworks like `Kubeflow` or `Argo Workflows` for managing ML tasks.
- **Kubernetes for ML:** Understanding basic Kubernetes architecture (Pods, Services, Deployments) and how it scales ML workloads dynamically.

Week 12: Capstone Project & Career Readiness

- **MLOps System Design:** Practicing architectural whiteboard sessions, designing end-to-end scalable ML systems for enterprise scenarios.
- **Technical CV Optimization:** Highlighting cloud platforms, CI/CD tools, and containerization skills effectively on resumes and LinkedIn.
- **Interview Preparation:** Simulating technical interviews focusing on model deployment challenges, debugging production ML systems, and behavioral questions.

[Project 2: End-to-End Automated MLOps Pipeline (Capstone)]

- **Scope:** Implement a fully automated MLOps pipeline where a GitHub code commit triggers automated data pulling via DVC, model retraining and tracking via MLflow, CI/CD deployment via GitHub Actions, and live health/drift monitoring via Grafana dashboards.