NIRMA UNIVERSITY

Institute:	Institute of Technology, School of Technology
Name of Programme:	MTech CSE, MTech CSE (Data Science)
Course Code:	6CS374ME25
Course Title:	MLOps
Course Type:	Department Elective-I
Year of Introduction:	2025-26

L	T	Practical Component				C
		LPW	PW	W	S	
3	0	2	-	-	-	4

Course Learning Outcomes (CLOs):

At the end of the course, students will be able to:

1. apply version control techniques to manage machine learning code and models (BL3)
2. model data pipelines and feature engineering workflows for machine learning projects (BL3)
3. analyse machine learning models in production environments, considering factors like scalability, reliability, and performance, and monitor and evaluate machine learning models in production to ensure their continued effectiveness (BL4)
4. build scalable and reproducible machine learning pipelines using containerisation. (BL6)

Unit	Contents	Teaching Hours (Total 45)
Unit-I	**Introduction to MLOps:** Definition and goals of MLOps, Challenges in deploying and managing machine learning models, Overview of the MLOps life cycle, Introduction to infrastructure virtualisation and containerisation	04
Unit-II	**Version Control for Machine Learning:** Introduction to Git and version control systems, Branching and merging strategies for collaborative ML development, Managing data and model versioning	04
Unit-III	**Data Pipelines and Feature Engineering:** Data preprocessing techniques for ML projects, building data pipelines using tools like Apache Airflow or Kubeflow Pipelines, Feature engineering best practices	07
Unit-IV	**Containerisation for ML Deployment: I**ntroduction to Docker and containerisation concepts, Building containerised ML applications, and Orchestration with Kubernetes for scalable deployments.	07
Unit-V	**Deploying ML Models:** Deployment options: cloud, on-premises, edge devices, Infrastructure considerations for model serving, Message Passing Infrastructure, Strategies for managing model versions and A/B testing	07

Unit-VI	**Monitoring and Evaluation:** Monitoring model performance and drift, Logging and metrics for ML systems, Evaluating model fairness and bias	06
Unit-VII	**Model Retraining and CI/CD:** Strategies for model retraining and updating, Continuous integration and continuous deployment (CI/CD) in MLOps, Test automation and quality assurance	07
Unit-VIII	**Ethical Considerations in MLOps:** Privacy and data protection in ML systems, Ethical considerations in model deployment and usage, Fairness, transparency, and accountability in MLOps.	03

Self-Study:
The self-study contents will be declared at the commencement of the semester. Around 10% of the questions will be asked from self-study content.

Suggested Readings/ References:

1. Emmanuel Ameisen, Building Machine Learning Powered Applications: Going from Idea to Product, O'Reilly
2. Foster Provost and Tom Fawcett, Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking, O'Reilly
3. Trevor Grant, Holden Karau, Boris Lublinsky, Richard Liu, and Ilan Filonenko, Kubeflow for Machine Learning: From Lab to Production, O'Reilly
4. Mark Treveil, MLOps: Continuous Delivery and Automation Pipelines in Machine Learning, Packt Publishing
5. Chris Fregly and Antje Barth, Hands-On MLOps: Continuous Integration and Deployment for Machine Learning, O'Reilly.

Suggested List of Experiments:

Sr. No.	**Name of Experiments/Exercises**	**Hours**
1	Version Control with Git • Create a GitHub repository. • Implement version control for a simple Python script or Jupyter Notebook used for data preprocessing. • Collaborate with a partner to demonstrate the use of branching and merging in Git.	04
2	Data Pipeline Creation: • Collect a dataset of your choice (e.g., CSV, JSON, or SQL data). • Create a data preprocessing pipeline using libraries like pandas and scikit-learn. • Automate the data ingestion and preprocessing steps	04
3	Feature Engineering • Select a real-world dataset and identify potential features. • Implement feature engineering techniques such as one-hot encoding, feature scaling, and feature selection. • Compare model performance before and after feature engineering	02
4	Containerization with Docker • Dockerize a machine learning model and its dependencies.	04

- Create a Dockerfile and Docker Compose file for a simple Flask web application that serves the model.
- Run and test the containerized application locally

5 Model Deployment 04

- Deploy a machine learning model on a cloud platform like AWS, Azure, or Google Cloud.
- Set up a REST API endpoint for the deployed model.
- Secure the endpoint and control access using authentication and authorization

6 Scalable Model Deployment 02

- Deploy a machine learning model using a serverless architecture (e.g., AWS Lambda, Azure Functions).
- Configure auto-scaling based on incoming traffic to ensure the application can handle varying workloads efficiently

7 Monitoring and Evaluation 02

- Implement model monitoring by setting up alerts for performance metrics (e.g., accuracy, latency).
- Use monitoring tools like Prometheus and Grafana to visualize and track model performance in real-time

8 Model Retraining and Updates: 02

- Develop a strategy for automated model retraining based on incoming data.
- Implement an update mechanism allowing easy deployment of model updates without downtime

9 CI/CD Pipeline: 04

- Create a CI/CD pipeline for your machine learning project using tools like Jenkins, Travis CI, or GitHub Actions.
- Automate the testing, building, and deployment processes for your project

10 Continuous Integration for Model Training: 02

- Set up a continuous integration pipeline that automatically retrains and updates a machine-learning model when new data becomes available.
- Use a tool like Jenkins or GitLab CI/CD to automate the retraining process and push updated models to production.