Course Title: MLOps in Python: From Development to Deployment

Instructor: Stephen Fickas

Contact Information: stephenfickas@gmail.com

Course Format: Remote, Asynchronous, with Interactive Support. While the core learning is through pre-recorded videos and Jupyter Notebooks, the instructor will foster an interactive learning environment through:

- Dedicated email support for timely responses to student inquiries.
- Scheduled virtual office hours using Zoom for personalized assistance and discussions.
- Opportunities for collaborative learning and discussion through online platforms.

Prerequisites: Solid Python programming skills, including object-oriented programming (OOP).

Course Description: This course provides a comprehensive introduction to MLOps, the practice of applying DevOps principles to machine learning systems. Students will learn how to build, deploy, and explain machine learning models in production using Python and industry-standard tools. Through hands-on exercises and practical examples, students will gain the skills necessary to automate and scale machine learning workflows.

Course Objectives: Upon successful completion of this course, students will be able to:

- Understand the core principles and practices of MLOps for supervised-learning and tabular data.
- Design and implement automated machine learning pipelines.
- Apply best practices for testing and validating machine learning models.
- Develop end-user documentation that explains design choices and model results (explainable AI) delivered during production.
- Deploy machine learning models to a web site that supports live demos.

Required Materials:

- Access to a computer with an internet connection.
- Google account.
- Chapters (Jupyter Notebooks) for the course (provided).
- Videos for each chapter (provided).

Course Structure:

First session (chapters 1-6): Wrangling.

- Feature Engineering.
  - Dropping/Adding features.
  - Downsampling/Upsampling rows.
- Basic wrangling components: Transformers and Pipelines.
  - Transforming categorical columns.
  - Handling outliers.

    - Scaling.
    - Imputation.

Second session (chapter 7): Transition to machine learning.

- Introduction to the Train-Test split.
- Causes and cures for data leakage.

Midterm 1 (chapters 1-7)

Third session: (chapters 8-12): Introduction to a representative set of machine learning models.

- Logistic Regression.
- KNN.
- Random Forests and Boosting.
- Artificial Neural Nets.

Fourth session (chapter 13): Explainable AI

- Feature importance with LIME.
- Thresholds and metrics.
- Wrangling documentation of design choices.

Midterm 2 (chapters 1-13)

Fifth session (chapters 14-15): Model tuning

- Tuning metrics (Precision, Recall, F1, AUC)
- Tuning algorithms (Grid, Random, Halving)

Final Project

- Students select their own tabular dataset.
- Construction of Transformers and Pipeline including documentation of design choices.
- Model building and tuning using all four models.
- Capture of explainable AI components.
- Custom web-site in Flask.
    - Backend: load tuned models, pipeline and explainable AI components.
    - Frontend: Collect row/sample data from user.
    - Backend: Transform user data and gather each model's predictions.
    - Frontend: Display prediction results along with LIME explanation.
- Deploy using ngrok (or Cloud-based service) for live demos.

End one-quarter (10-week) course

Sixth session (chapters 16-20): pivot to image-analysis

- In essence, revisit earlier chapter topics but now applied to image-analysis. Introduces new wrangling steps (image preprocessing, augmentation), new models (CNN, InceptionV3, ViT DETR), and new explanation approaches (Grad-CAM, SHAP).

Final Project 2

- Students select their own image dataset.
- Construction of preprocessors.
- Model building and tuning using all image models.
- Capture of explainable AI components.

- Custom web-site in Flask.
  - Backend: load tuned models, pipeline and explainable AI components.
  - Frontend: Upload image from user.
  - Backend: Transform user image and gather each model's classification.
  - Frontend: Display classification results along with explanation.
- Deploy using ngrok (or Cloud-based service) for live demos.

End of one-semester (15 week) course