ngee ann
school of infocomm technology

Machine Learning 2

Diploma in Data Science (DS)
Oct 2025 Semester

INDIVIDUAL ASSIGNMENT 1

(40% of Machine Learning Module)

Deadline for Submission:

8th Feb 2026 (Sunday), 2359 Hours

Student Name :	
Student Number :	

Penalty for late submission:

10% of the marks will be deducted every day after the deadline.
NO submission will be accepted after **15th Feb 2026, 23:59.**

Machine Learning Assignment 1

1. Assignment Overview

This assignment focuses on the application of **MLOps principles** to a real-world regression problem using the **Bike Sharing Demand (Daily) dataset**. You will build, evaluate, and operationalise machine learning (ML) models while demonstrating how data drift and automation are managed throughout the machine learning lifecycle.

This assignment requires you to:

- design and justify experiments,
- analyse the **impact of data drift on model performance**, and
- implement **automation and quality controls** using MLOps tools.

2. Dataset Description

2.1. Background

Bike-sharing systems represent a new generation of traditional bicycle rental services, in which the entire process—from membership registration to bicycle rental and return—is fully automated. These systems allow users to conveniently rent a bicycle from one location and return it to a different location.

2.2. Data Set

The bike-sharing rental process is strongly influenced by environmental and seasonal factors. For example, weather conditions, precipitation levels, day of the week, season, and hour of the day can significantly affect rental behavior.

You are provided with 2 sets of bike-sharing daily rental dataset for the year 2011 and 2012, named respectively as day_2011.csv and day_2012.csv. These datasets are collected from the Capital Bikeshare system in Washington, D.C., USA. This dataset shows the bike sharing counts aggregated on daily basis.

The data dictionary for the 2 datasets are given as follows:

Field Name	Description
dteday	Date
season	Season (1 = spring, 2 = summer, 3 = fall, 4 = winter)
mnth	Month (1 to 12)
holiday	Whether the day is a holiday (1 = holiday, 0 = not holiday)
weekday	Day of the week
workingday	1 if the day is neither weekend nor holiday, otherwise 0

weathersit	Weather situation: 1 = Clear / Few clouds / Partly cloudy 2 = Mist + Cloudy / Mist + Broken clouds / Mist + Few clouds 3 = Light Snow / Light Rain + Thunderstorm + Scattered clouds 4 = Heavy Rain + Ice Pallets + Thunderstorm + Mist / Snow + Fog
temp	Normalized temperature in Celsius (divided by 41)
atemp	Normalized feeling temperature in Celsius (divided by 50)
hum	Normalized humidity (divided by 100)
windspeed	Normalized wind speed (divided by 67)
cnt (target)	Total count of rental bikes

3. Description of Required Task

This assignment consists of a total of 3 tasks, corresponding to key stages of the ML and MLOps lifecycle.

3.1. TASK 1 — Model Development and Experiment Design

You are required to develop and evaluate machine learning models for predicting daily bike rental demand **using only "day_2011.csv" dataset** through experiment design. This task focuses on model development and experimentation.

It is recommended to perform the following steps for Task 1.

Steps	Details
Project Scoping	Briefly info of the project scope
Data Preprocessing	Perform any necessary data preprocessing on the dataset. Some hints on data preprocessing: • If using `dteday`, you must extract numeric features (e.g., month/weekday) and drop raw date
Experiment Design (Not Just Model Training)	You are required to actively design and conduct experiments to develop a suitable prediction model using **day_2011.csv**. Specifically, you must: • Use Linear Regression to develop the **baseline model.** • Develop **one improvement strategy** such as tree-based model with depth constraints or regularised regression For **each experiment**, explain: • **Why** it is performed • **What improvement or trade-off** is expected
Model Training, Evaluation, and Comparison	Using **day_2011.csv**, you must: • Train and evaluate all models • Use appropriate regression metrics (e.g. RMSE, MAE, R^2) • Compare results across experiments

MLflow Experiment Tracking	Use **Mlflow** to: • Track **at least two experiment runs:** o Baseline model o Improved model • Log: o Model type o Hyperparameters o Evaluation metrics (e.g. RMSE, MAE, R^2) o Model artifacts • Use Mlflow to compare results across experiments.
Model Selection and Registration	• Select the most suitable model based on: o Performance metrics o Model complexity o Expected robustness to changing data • Register the selected model in Mlflow. Export the selected Mlflow-registered model to a file (joblib/pkl) for later Task 3 testing. • Justify your choice clearly.

3.2. TASK 2 — Data Drift Analysis and Impact Assessment

Analyse **temporal data drift** by evaluating how a model trained on **day_2011.csv** performs when applied to **day_2012.csv** and recommend appropriate operational actions.

You are recommended to perform the following steps for Task 2.

Steps	Details
Data Drift Analysis	Compare **feature distributions** between the datasets day_2011.csv and day_2012.csv and perform the following: • Use **simple descriptive statistics**, such as: o Mean o Standard deviation • Include **at least one visualisation** (e.g. histogram or boxplot). Clearly identify: • Which features show noticeable drift • How the above-mentioned features contribute to the observed drift.
Impact of Drift on Model Performance	Using the selected model from Task 1: • Evaluate model performance on: o day_2011.csv o day_2012.csv (drifted data) • Compare metrics such as RMSE and MAE. Explain: • Whether performance degradation is observed when moving from Year 2011 to Year 2012 data set. • Which drifting features are most likely contributing to the degradation.

Drift Response and Operational Decision	Based on your analysis, recommend **one operational action**, such as: • Retraining the model using day_2012.csv data • Monitoring the model without retraining • Introducing separate models for different years or seasons • Increasing evaluation frequency Your recommendation must be supported by: • Evidence from drift analysis • Changes in model performance

3.3. TASK 3 — Automation using GitHub Actions

Automate model quality checks using a CI/CD pipeline to ensure only high-performing models pass the "Quality Gate." i.e. the **pass/fail checkpoint** that automatically checks whether your code/model meets the **minimum quality requirements as stated in assert or conditional statements,** before it is accepted in the workflow.

You are recommended to perform the following steps for Task 3.

Steps	Details
GitHub Repository Setup	Create a GitHub repository following the hierarchy as listed below. Your repository must include: • **src/ folder**: Containing your core Python scripts (scripts or notebooks used for Tasks 1 and 2) • **tests/ folder**: Containing a test script named test_model.py. • **data/ folder**: Containing the dataset (or a sample) used for evaluation. • **requirements.txt**: Listing the libraries needed (e.g., pandas, scikit-learn, mlflow). • **README.md** explaining how to run the project • **Saved Model**: Your best model file (e.g., model.pkl or model.joblib) from Task 1.
GitHub Actions Workflow	Set up the automated pipeline using the "Python application" template in GitHub Actions (as practiced in class): • Create the workflow file: .github/workflows/python-app.yml. • Configure the workflow to trigger automatically on a **git push** model to the main branch. The workflow should set up the Python environment, install dependencies from requirements.txt, and execute your test script.
Automated Quality Gate	In your tests/test_model.py script, implement a "Quality Gate" check. This script should:

(The "Test" Script)	• Load the saved best model from task 1 and the evaluation data. • Calculate the model's performance (e.g., RMSE). • Implement the Gate using an assert statement or a conditional sys.exit(1) to terminate the workflow with a failure status if the performance of the saved model does not meet the threshold (e.g., assert: rmse <= 0.95 * rmse_baseline) *Note: If the condition is not met, the GitHub Action must show a failed (red) status.*
Evidence of Automation	Provide the following evidence in your report: Note: All **snapshots** must clearly show the GitHub **repository owner/username** (e.g., in the URL bar or repository header) to verify ownership. • A snapshot of the GitHub repository setup done in the first step of this task. • A snapshot of your GitHub Actions tab, showing a **pass/fail (green/red)** workflow run. • A snapshot of the workflow logs showing your `test_model.py` script executing successfully.

4. Deliverables

For this assignment, you must submit **all** of the required deliverables punctually.

4.1. Jupyter Notebook [Deadline: **8th Feb 2026 (Sunday), 2359 hrs**]

- The codes for all the 3 tasks should be completed in a single jupyter notebook file with necessary comments and markdowns.
- Rename your notebook as **ML2_ASG_notebook_<name>.ipynb**
- Submit via the designated submission folder in POLITEMall
- Ensure that your code is well-commented using Markdown

4.2. GitHub zipped folder [Deadline: **8th Feb 2026 (Sunday), 2359 hrs**]

All the files related to task 3 execution (test_model.py and ymal file, requirements.txt) will be submitted as zipped folder named **ML2_ASG_GitHub_<name>.zip.**

4.3. Video Recorded Presentation [Deadline: **8th Feb 2026 (Sunday), 2359 hrs**]

- You are required to submit individually an online presentation through **Bongo**. The maximum video duration is **10 minutes**. Focus on experiment rationale, drift analysis and MLOps decisions
- Detailed instructions on using Bongo are given in Appendix 1.
- You are encouraged to use your Jupyter Notebook for the presentation

- Presentations exceeding the time limit will be penalised

4.4. Report [Deadline: **8th Feb 2026 (Sunday), 2359 hrs**]

- You are required to submit an individual report (**PDF or Word document**) named as **ML2_ASG_Report_<name>.pdf** or **ML2_ASG_Report_<name>.doc**, containing the sections listed in **Section 5** below.
- You are to follow the suggested format of the report as given in Section 5. The report should focus on **interpretation, justification, and decision-making**, not just code output.

4.5. Question and Answer (Q&A) Session

- You must attend a **10-minute face-to-face Q&A session**
- You are to arrange for a face-to-face Question-and-Answer session with your tutor.
- You will be required to respond to all technical questions posed by your tutor regarding your submissions.
- You will be assessed on your conceptual understanding, ability to justify design choices and interpretation of drift and automation results.

Submit the deliverables no later than **Sunday 8 Feb 2026, 23.59 hours** in POLITEMall. Late submissions of assignment-based coursework component without leave of absence (LOA) for the module will be subjected to the late penalty.

Note:

- **DO NOT make your GitHub Repository setting as "public"**
- **DO NOT PLAGIARIZE (please refer to Ngee Ann Polytechnic Plagiarism Policy webpage for more information)**

5. Suggested Report Format and Content Guidelines

Submit an individual report in PDF or Word format named as **ML2_ASG_Report_<name>.pdf** or **ML2_ASG_Report_<name>.doc,** with the following sections (see table below). Sample content description is provided for each section. You are free to include other relevant information you deem necessary in the sections. You are strongly encouraged to include screen shots in your explanation, description and/or analysis.

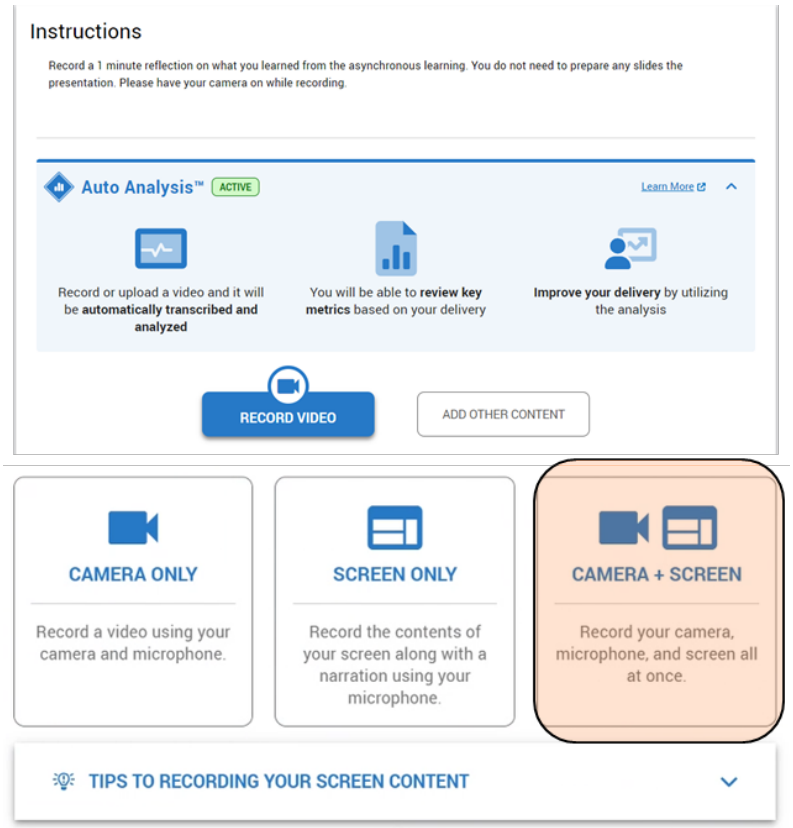
Section	**Content Guidelines**	**Word Count**
Table of Contents		NA
Overview	• Project objectives • Dataset overview (Bike Sharing – Daily) • Brief summary of tasks and key outcomes	Min: **200 words** Max: **500** words
Task 1: Model Development and Experiment Design (Bike Sharing Demand Prediction)	• **Data Preprocessing** • **Experiments and Results** • **MLflow Evidence:** Show 2 runs (baseline vs improved) in MLflow (screenshot/summary) • **Model Selection with justification**	Min: **300 words** Max: **800 words**
Task 2: Data Drift Analysis and Impact Assessment	• **Data Drift Analysis** • **Impact of Drift on Model Performance** • **Drift Response and Recommended action with justification**	Min: 3**00 words** Max: 8**00 words**
Task 3: Automation and Quality Control using GitHub Actions	• **GitHub Repository Structure** • **GitHub Actions Workflow:** Describe what triggers the workflow and the steps the workflow performs. • **Quality Check Implementation** o Explain the logic inside your test_model.py o Include screenshots (including capture of owner/username on top left corner) of the GitHub Actions logs showing a pass/fail **(green/red)** depending on if the model met the threshold. • **Role of CI/CD in MLOps** o Short discussion: How does this automated test protect the production environment.	Min: 3**00 words** Max: 8**00 words**
Summary, Reflection, and Further Improvements	• **Overall Summary** o Key findings from Tasks 1–3 • **Further Improvements** o Possible improvements to modelling, drift handling, or automation • **Reflection** o Reflection on learning outcomes achieved o Brief evaluation of the use of Generative AI tools (if any) in the learning.	Min: 2**00 words** Max: 5**00 words**

1. Grading Criteria

Component	Assessment Criteria	Weightage
Jupyter Notebook & Task 3 related codes	a) Completeness of code for the 3 tasks b) Quality of code c) Quality of Markdown comments d) Evidence of MLflow experiment tracking logs and model registration within the notebook	**40%**
Final Report	a) Quality of report b) Completeness based on report guidelines c) Justification of modelling and MLOps decisions d) Clarity, structure, grammar, and use of visual aids e) Quality of drift response and improvement recommendations and quality of CI/CD 'Quality Gate' justification.	**30%**
Online Video Presentation	a) Quality of work b) Logical flow based on content guidelines c) Clarity of explanation and visual support d) Presentation and articulation skills. e) Completeness of presentation including the GitHub Actions run Green/Red (pass/ fail) + MLflow tracking/registry.	**20%**
Face-to-Face Q&A	a) Clear understanding of approach b) Justification of modeling and MLOps decisions c) Interpretation of results using evidence d) Conceptual knowledge of ML and MLOps	**10%**
Total		**100%**

Appendix I: Instructions on Video Recorded Presentation

- You are required to do an **online presentation** and share your findings. The presentation **should not exceed 10 minutes**. The presentations which exceed the allotted time will be penalized.
- Students will make use of the video assignment app, powered by Bongo, to capture their presentations. Each student is to practice the presentation in advance to ensure completion **within 10 minutes**. The recording must include both webcam (clearly showing the student's face for authentication) and slides or codes (whichever is applicable).
- Select the **RECORD VIDEO** option and choose **CAMERA + SCREEN** as shown in the figure below. The figure may differ with the constantly update of the Bongo software, hence students may see a different layout but general steps should still apply.



Instructions

Record a 1 minute reflection on what you learned from the asynchronous learning. You do not need to prepare any slides the presentation. Please have your camera on while recording.

Auto Analysis™ ACTIVE

Record or upload a video and it will be **automatically transcribed and analyzed**

You will be able to **review key metrics** based on your delivery

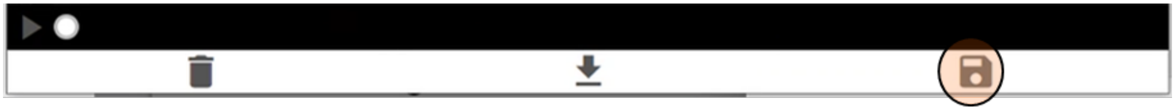
Improve your delivery by utilizing the analysis

Record a video using your camera and microphone.

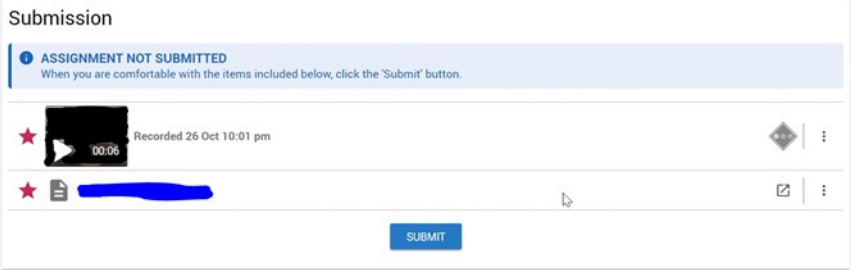
Record the contents of your screen along with a narration using your microphone.

Record your camera, microphone, and screen all at once.

- After recording the video, click save (as shown below) and it will be ready for students to append it for submission.



- Select the video by clicking on the Star and click **SUBMIT**.



Submission

ASSIGNMENT NOT SUBMITTED
When you are comfortable with the items included below, click the 'Submit' button.

Recorded 26 Oct 10:01 pm

[redacted]