# Uganda Christian University (UCU)

## Object Oriented Programming (OOP)

**Course Level:** Year 1, Semester 1

**Course Code :** CSC8101

**Semester:** Advent *2025*

| Full Name | Registration Number | Access Number |
|---|---|---|
| OGWAL Pius | S25M19/012 | B34982 |

**Lecturer:** Dr. John Habere

**Date: 14th October 2025.**

**Assignment 2:**
**Object Oriented Python Programming – Project Proposal Assignment**

**Topic: Machine Learning-based Predictive Derisking for Dairy Investment in Nakasongola District.**

## 1.1 Background

Nakasongola District sits within Uganda's cattle corridor, where dairy enterprises are highly exposed to rainfall variability, heat stress, feed price shocks, electricity outages, and volatile farm-gate milk prices. Investment decisions by cooperatives, lenders, and farmers often rely on anecdotal judgment rather than structured evidence, increasing the risk of mistimed or suboptimal capital allocation. Recent advances in Python-based machine learning make it feasible to transform routinely available administrative records (milk output, costs, vet visits, loan access) and open climate proxies (CHIRPS rainfall, NASA POWER temperature, MODIS NDVI) into timely, explainable risk signals. An object-oriented system that ingests these indicators, learns patterns from historical months, and produces a monthly "investment feasibility" score with short explanations can help de-risk decisions while remaining lightweight and transparent for low-data contexts like Nakasongola Cattle Cooridor.

## 1.2 Problem Statement

Private dairy investors in Uganda's cattle corridor face uncertainty in production, climate and market conditions. Decisions rely on anecdote, leading to underinvestment or mistimed capital. This project proposes an object-oriented python system that ingests multi-source indicators and predicts monthly investment feasibility, with concise risk explanations for decision support.

## 1.3 Purpose:

To design a Python oriented prototype that ingests de-identified Nakasongola farm/cooperative and open climate data(Jan2023–Dec2024), trains baseline machine learning models, and outputs interpretable monthly investment-feasibility scores to de-risk dairy decisions.

### 1.3.1 Specific Objectives

1. To design and implement an OOP Python pipeline for dairy investment risk prediction.
2. To train and validate baseline predictive models using retrospective panel data (Jan2023–Deck2024).
3. To deliver actionable, explainable outputs for decision support.

## 1.4 Relevance in real-world context

Dairy is an essential investment for capital in Nakasongola District yet sensitive to rainfall variability, heat stress, and price volatility. A small, explainable model that converts available indicators into an investment-feasibility signal can de-risk capital for lenders, cooperatives and farmers.

## 1.5 Scope

Build a modular object-oriented programming using python (OOP) system with:

a. Data models and validation
b. Pipelines for loading, cleaning and features
c. Baseline classifiers (Logistic Regression and Random Forest)
d. Explainability (feature importances/SHAP); and,
e. Interfaces: Command-Line Interface (CLI), and an interactive dashboard(streamlit) for visualizing predictions, model performance, and drivers.
f. Deliver clean code, tests, Nakasongola district dataset (Jan 2014–Dec 2024), and a README.

## 1.6 Expected Outcomes and Deliverables

- Trained baseline model with accuracy/ROC-AUC and confusion matrix.
- Global feature importances and example local explanations.
- Explainable, actionable feasibility scores for decision support
- Auto-generated feasibility report (DOCX/CSV) summarizing predictions by month/farm.

## 1.7 Approach & Methodology (Data Acquisition & Modelling)

The study will adopt a retrospective panel data (Jan 2023–Dec 2024) for Nakasongola (District) Cattle Corridor, combining de-identified farm/cooperative monthly records (milk output, feed costs, vet visits, outages, loan access, farm-gate prices), public climate proxies (monthly rainfall, temperature, NDVI aggregated to sub-county), and simple program logs (e.g., loan_access). Working with existing data minimizes

burden/ethics risks, captures both wet/dry seasons, and enables a clean temporal validation (train on 2023 to predict 2024).
The target investment_feasible (0/1) will be derived from transparent business rules (positive margin after feed costs within a price band and limited outages). Predictors include milk_liters, feed_cost_ugx, vet_visits, loan_access, outages_days, market_price_ugx_per_liter, temp_c, rainfall_mm, ndvi, plus lagged/seasonal terms crafted to avoid leakage.

Data will be schema-validated, type-cast, winsorized (1st to 99$^{th}$ pct), and imputed (median within farm-season) with flags; derived ratios are normalized. Models: Logistic Regression and Random Forest as baselines, tuned via 5-fold CV on the 2023 train set, then evaluated out-of-time on 2024 using ROC-AUC, F1, and confusion matrices. Overfitting will be controlled through compact features, regularization (LogReg) and tree constraints (RF). Explainability will be provided through global feature importances and, if needed, SHAP examples. The OOP design wraps this into four classes: DataSource (load/merge/validate), FeatureBuilder (lags/impute/scale), RiskModel (train/evaluate/persist), and DeRiskingReport (month-by-month feasibility with brief explanations).Tools and approaches will involve:

OOP classes: DataSource, FeatureBuilder, RiskModel, DeRisking Report; Libraries: pandas, numpy, scikit-learn; shap; python-docx for reporting; Methods: train/test split, cross-validation, confusion matrix, ROC-AUC, threshold tuning; and Deliverables: GitHub repo, dataset (CSV), 2-page proposal, and README.

### 1.8 Data (Nakasongola, Jan 2023–Dec 2024)

Dataset will include climate proxies monitored against (temperature, rainfall, NDVI), production (milk), prices/costs, outages, finance access. The target label will include: investment_feasible (0/1). Climate and environmental inputs will be sourced from open datasets—CHIRPS (rainfall), NASA POWER (temperature/precipitation), and MODIS NDVI aggregated via Google Earth Engine while production, financial, and outage variables will come from de-identified cooperative/farm administrative logs in Nakasongola District (Jan2023–Dec2024).

### 1.9 Timeline & Risks

#### 1.9.1 Timeline

**Table 1: Timeline (4 weeks)**

| Week | Focus | Key Tasks/Deliverables |
|---|---|---|
| 20/Oct | Design & Dataset | Finalize problem scope; collect/clean Nakasongola dataset; define features/labels; EDA brief |
| 27/Oct | Pipeline & Baselines | Build modular OOP pipeline (ingest to prep to model); implement baseline models; baseline report |
| 3/Nov | Tuning & Explainability | Hyperparameter tuning (CV); add SHAP/feature importances; validation results |
| 10/Nov | Packaging & Demo | Package as CLI/notebook; create demo & README; finalize doc and slides. |

#### 1.9.2 Key Risks & Mitigations

**Table 2: Key Risks and Mitigations**

| Risk | Potential Impact | Mitigation Strategy |
|---|---|---|
| Data sparsity/ noise | Weak/generalizable models | Use parsimonious features; robust preprocessing; k-fold CV |
| Model opacity | Low stakeholder trust | Feature importances/SHAP plots; clear model cards |
| Overfitting | Poor real-world performance | Compact feature set; regularization; train/val/holdout split |

### References

Paudyal & Kafuko (2024); Kahiu et al. (2024); Nalukwago et al. (2023); Espinoza-Sandoval et al. (2024)