Part 1: Short Answer Questions (30 points)

1. Problem Definition (6 points)

- Define a hypothetical AI problem (e.g., "Predicting student dropout rates").
  Predicting crop disease outbreaks in Kenyan smallholder farms 2 – 4 weeks in advance.

- List **3 objectives** and **2 stakeholders**.
  Objectives:
  1. Achieve ≥ 85% recall on disease-positive fields to enable early intervention.
  2. Reduce yield loss by at least 20% in participating counties through timely alerts.
  3. Deliver offline-capable predictions in Swahili and English via PWA.
  Stakeholders:
  a. County agricultural extension officers
  b. Seed/chemical input suppliers *(e.g. Kenya Seed Company)*.

- Propose **1 Key Performance Indicator (KPI)** to measure success.
  F2-score on the test set → prioritizes recall over precision while balancing both.

2. Data Collection & Preprocessing (8 points)

- Identify **2 data sources** for your problem.
  Data Sources:
  1. Satellite imagery (Sentinel-2 NDVI & moisture indices) + weather API (temperature, rainfall, humidity).
  2. Crowdsourced farmer reports via mobile app/USSD + historical KALRO disease incidence records.

- Explain **1 potential bias** in the data.
  Crowdsourced reports are higher in counties with better mobile network → under-represents remote arid/semi-arid areas.

- Outline **3 preprocessing steps** (e.g., handling missing data, normalization).
  1. Cloud masking and 10-day composite creation for Sentinel-2 imagery.
  2. Resample all time-series data to 10-day intervals and create lagged features (NDVI t-1, t-2, rainfall cumulative last 30 days).
  3. SMOTE + Tomek links to handle severe class imbalance (disease outbreaks = 5–8% of samples).

3. Model Development (8 points)

- Choose a model (e.g., Random Forest, Neural Network) and justify your choice.
  LightGBM with temporal features – justified by fast training on tabular + time-series data, handles missing satellite data natively, excellent performance on imbalanced agricultural

datasets, and low inference latency for mobile deployment.

- Describe how you would split data into training/validation/test sets.
  Temporal split → training (2018–2021), validation (2022), test (2023–2024) to avoid leakage across seasons.

- Name **2 hyperparameters** you would tune and why.
  a. num_leaves (controls model complexity → prevent overfitting on noisy satellite data).
  b. min_child_samples (increases robustness when disease events are rare).

4. Evaluation & Deployment (8 points)

- Select **2 evaluation metrics** and explain their relevance.
  1. Recall at 80% precision → critical to catch real outbreaks; false positives are cheaper than missed ones.
  2. PR-AUC → better than ROC-AUC with extreme class imbalance.

- What is **concept drift**? How would you monitor it post-deployment?
  Concept drift: Changes in climate patterns or new virus strains.
  How to monitor post-deployment: Monitor weekly using Kolmogorov-Smirnov test on feature distributions and alert if model recall drops > 15% on rolling 30-day window; trigger retraining pipeline.

- Describe **1 technical challenge** during deployment (e.g., scalability).
  Satellite data latency and resolution mismatch → Sentinel-2 has a 5 - 12 day revisit time in cloudy equatorial regions, causing delayed or missing inputs during critical outbreak windows.
  <u>Solution:</u> Fuse with daily MODIS (lower resolution) + harmonized Landsat-Sentinel dataset; use a small imputation model (trained with Prophet + weather proxies) to fill gaps in real-time before feeding into the main LightGBM model.

MELISSA