Part 1: Short Answer Questions (30 points)

1. Problem Definition (6 points)

Hypothetical AI Problem:

Predicting crop disease outbreaks in Kenyan maize farms using satellite and sensor data.

Objectives:

• Detect early signs of crop disease from environmental and image data.

• Alert farmers and agricultural officers for timely intervention.

• Reduce crop losses and improve food security.

Stakeholders:

• Smallholder farmers in Kenya.

• Ministry of Agriculture and ICT Authority.

Key Performance Indicator (KPI):

• Disease detection accuracy (%) — proportion of correctly identified disease cases vs total cases.

2. Data Collection & preprocessing (8 points)

Data Sources:

• Satellite imagery (e.g., NDVI, temperature, humidity).

• IoT sensor data from soil and weather monitoring devices.

Potential Bias:

• Geographic bias: Data may be concentrated in well-funded regions, underrepresenting remote or low-resource farms.

Preprocessing Steps:

• Handle missing sensor readings using interpolation or imputation.

• Normalize environmental variables (e.g., temperature, moisture) for consistent scale.

• Augment image data (e.g., rotation, zoom) to improve model robustness.

3. Model Development (8 points)

Chosen Model:

Convolutional Neural Network (CNN) — ideal for analyzing spatial patterns in satellite and image data.

Data Splitting Strategy:

• 70% training, 15% validation, 15% test — stratified to maintain class balance (healthy vs diseased crops).

Hyperparameters to Tune:

• Learning rate: Controls how fast the model updates weights — critical for convergence.

• Number of convolutional layers: Affects model depth and ability to capture complex patterns.

4. Evaluation & Deployment (8 points)

Evaluation Metrics:

• Precision: Measures how many predicted disease cases were correct — important to avoid false alarms.

• Recall: Measures how many actual disease cases were detected — crucial for timely intervention.

Concept Drift:

• Occurs when the relationship between input data and target changes over time (e.g., new disease strains or climate shifts).

• Monitor using periodic re-evaluation on recent data and tracking drops in model accuracy.

Technical Challenge:

• Scalability: Deploying the model across diverse regions with limited connectivity and hardware — requires lightweight models and edge computing solutions.

Part 2 — Case Study: 30-day readmission risk predictor

1) Problem scope (5 pts)

Problem definition. Predict whether a discharged patient will be readmitted to the hospital within 30 days (binary classification: *high risk* / *low risk*).
Objectives.

Identify high-risk patients before discharge so care teams can intervene (e.g., discharge planning, follow-up calls, medication reconciliation).

Reduce preventable readmissions and associated costs while improving patient outcomes.

Provide interpretable risk drivers so clinicians can act.

Primary stakeholders.

Patients (benefit from better follow-up and fewer avoidable readmissions).

Clinicians (physicians, nurses, case managers) who will receive risk alerts and use the model's explanations.

Hospital administration / quality improvement teams (care redesign, reimbursement).

IT / EHR vendor (integration, data flows).

Privacy/compliance officers and legal (HIPAA/agreements).

---

2) Data strategy (10 pts)

Proposed data sources

Electronic Health Records (EHR): demographics, diagnoses (ICD), problem list, vital signs, labs, discharge disposition, clinical notes.

Hospital utilization / claims: prior admissions, ED visits, length of stay.

Pharmacy / medication fills: discharge meds, recent prescriptions.

Social determinants / external sources: socio-economic status proxies (neighborhood, insurance type), available care-access info.

Device / home health referrals / follow-up appointments.
Two ethical concerns

Patient privacy & data protection: Sensitive health data must be protected from unauthorized access or misuse.

Bias and fairness / disparate impact: Training data may underrepresent certain demographic groups (race, language, socioeconomic status) or reflect historical differences in care access, causing biased risk scores or unequal treatment. You must detect and mitigate biases to avoid worsening disparities

Preprocessing pipeline (stepwise)

a) Data ingestion & governance

Pull relevant EHR tables (encounters, diagnoses, meds, labs, social history) and link via patient ID. Log provenance and access.

b) Cleaning

Standardize timestamps, unify coding systems (ICD-9/10 mapping), handle duplicates.

c) Labeling

Define positive label = any inpatient readmission within 30 days of discharge. Use clearly documented logic.

d) Feature engineering (examples)

Static features: age, sex, insurance type, comorbidity counts (e.g., Charlson index), number of prior admissions in last 6/12 months.

Index admission features: length of stay, discharge disposition (home, SNF), primary diagnosis group, number of procedures.

Labs/vitals trends: last value before discharge and simple slopes (e.g., creatinine trend last 48–72 hrs).

Medication features: number of discharge meds, presence of high-risk meds (anticoagulants, opioids).

e) Encoding & scaling

One-hot or target encoding for categorical variables (careful with rare categories). Scale continuous variables if model benefits.

f) Missing data strategy

Create indicators for missingness (missingness can be informative). Impute with clinically sensible values or model-specific imputation (e.g., median or iterative imputer).

g) Train/validation/test split

Use temporally split data (train on earlier discharges, validate/test on later) to simulate prospective performance and avoid leakage. Also use stratified sampling for rare events.

h) Bias & fairness checks

Evaluate performance across subgroups (age, race, gender, insurance). Use fairness tools and mitigation if disparities are found.

i) Logging & versioning

Store dataset versions, feature definitions, and data lineage for reproducibility and audits.

3) Model development (10 pts)

Model selection & justification

Gradient-boosted trees (e.g., XGBoost)

Why: strong predictive performance on heterogeneous tabular EHR data, handles missingness and categorical variables well, and supports feature importance for explainability.

Hypothetical confusion matrix + precision/recall (10 pts)

Assume evaluation set = 1,000 discharged patients. Model predicts positive = will be readmitted within 30 days.

Confusion matrix (hypothetical):

True Positives (TP) = 80

False Positives (FP) = 40

False Negatives (FN) = 20

True Negatives (TN) = 860

Verify totals: 80 + 40 + 20 + 860 = 1000.

Precision: = TP / (TP + FP)

= 80 / (80 +40) = 80 / 120 = 1 / 3 = 0.66666… = 66.67%

Recall (Sensitivity): = TP / (TP + FN)

TP + FN = 80 + 20 = 100.

Recall = 80 ÷ 100 = 0.8 = 80.00%.

Accuracy = (TP + TN) / total = (80 + 860) / 1000 = 940 / 1000 = 0.94 → 94.0%.

F1 score = 2 × (precision × recall) / (precision + recall)

precision × recall = (2/3) × (4/5) = 8/15.

precision + recall = (2/3) + (4/5) = (10/15 + 12/15) = 22/15.

F1 = 2 × (8/15) ÷ (22/15) = (16/15) × (15/22) = 16/22 = 8/11 ≈ 0.7273 → 72.73%

4) Deployment (10 pts)

Steps to integrate into hospital system

a) Clinical problem fit & pilot design

Co-design with clinicians; choose use case (e.g., flag high-risk patients 24 hours before discharge). Define clinical actions per risk tier and governance.

b) Model packaging & registry

Containerize the model (Docker), register in a model registry (with versioning, metadata, and reproducible environment).

c) Create a secure inference API

Expose a narrow, authenticated REST/FHIR API that accepts only required inputs and returns risk + explanations.

d) EHR integration

Integrate via standard interfaces. Deploy as an EHR decision support service or middleware that writes a risk score to a dedicated EHR field or pushes an alert into clinician workflow.

e) Monitoring & feedback

Monitor model performance (AUC, calibration), data drift, and fairness metrics by subgroup in production. Capture outcomes to retrain/validate periodically. Log predictions and clinician responses (for auditing).

f) Governance & change control

Establish governance committee (clinicians, informatics, compliance) to approve model updates, SOPs for when to retrain, thresholds for alerts, incident response.

g) Testing & phased rollout

Start with silent/observational deployment, then a limited pilot, then broader rollout if validated clinically.

HIPAA & regulatory compliance (how to ensure)

Key safeguards and actions (practical):

Business Associate Agreements (BAAs) with any cloud / vendor handling PHI. Ensure contracts assign responsibilities.

Administrative safeguards: Risk assessments, policies, staff training, and incident response plans.

Data minimization: Model access only the fields required for the prediction; avoid unnecessary PHI.

Monitoring for breaches and reporting: Implement detection and breach notification workflows per HIPAA rules.

Clinical safety & validation: Validate model prospectively; document clinical validation and maintain versioned artifacts for audits. Guidance on combining AI and HIPAA is evolving — explicit documentation and BAAs are essential.

5) Optimization — one method to address overfitting (5 pts)

Method: Stratified k-fold cross-validation with early-stopping and regularization (for tree models)

How it works (concise):

Use stratified k-fold CV (e.g., k=5 or 10) on the training set to estimate out-of-sample performance and to tune hyperparameters.

This combination reduces overfitting by preventing the model from fitting noise (early stopping) and penalizing overly complex models (regularization), while stratified CV stabilizes estimates for an imbalanced target.

Why one method: it’s practical, directly supported in production libraries (XGBoost/LightGBM), and balances bias/variance for tabular HER data

Part 3: Critical Thinking (20 points)

1. Ethics & Bias (10 points)

How biased training data might affect patient outcomes

Biased training data can lead to unequal or harmful decisions for certain patient groups. Some examples:

a) Underrepresented groups receive inaccurate risk scores.

If minority patients, rural populations, or low-income patients appear less frequently in the dataset, the model may systematically underpredict their readmission risk, leading to fewer follow-up resources, poorer monitoring, and worse outcomes.

b) Historical healthcare inequalities become encoded into the model.

If past medical records reflect unequal access to treatments, then the model may "learn" that these patients are lower priority, reinforcing discrimination.

c) False positives burden certain groups.

Overpredicting readmission risk for some populations could cause unnecessary interventions, extra testing, or anxiety while wasting hospital resources.

d) Reduced trust in AI.

If some groups consistently receive incorrect or unfair predictions, this damages trust in the system and may discourage patients from seeking care.

e) False negatives: High-risk patients not flagged for follow-up, increasing chances of readmission or complications.

One strategy to mitigate this bias

Stratified Sampling & Fairness Audits: Ensure balanced representation across age, gender, ethnicity, and socioeconomic status during data collection. Regularly audit model performance across subgroups to detect and correct disparities.

2. Trade-offs (10 points)

Trade-off between interpretability and accuracy in healthcare

In healthcare:

a) Interpretable models (Logistic Regression, Decision Trees)

Pros: Easy for clinicians to understand, easier to justify to regulators, high transparency.

Cons: May be less accurate for complex medical patterns.

b) Highly accurate but less interpretable models (Deep Neural Networks)

Pros: Capture complex relationships and often produce higher predictive performance.

Cons: Harder to explain why a prediction was made, which can reduce clinician trust and complicate ethical/legal accountability.

Trade-off:

Healthcare often prioritizes interpretability due to patient safety and legal risks, but must balance it with the need for high accuracy to avoid missed diagnoses or incorrect predictions. Many hospitals choose a combination:

- use the more accurate model for predictions,
- provide explanations using tools like SHAP,
- And keep a simple baseline model for transparency.

Impact of limited computational resources on model choice

If the hospital has limited computational resources (older servers, restricted cloud use, no GPUs):

Lightweight models are preferred: Logistic Regression, Naive Bayes, Small Decision Trees

These models train quickly, require minimal memory, and are easy to deploy.

Heavy models may be impractical eg Deep learning models.

Such models need more RAM, CPU time, and may increase latency during prediction.

Result:

The hospital might choose a simpler, more efficient model to ensure: fast prediction times, lower maintenance cost, no strain on hospital IT infrastructure.

Part 4: Reflection & Workflow Diagram (10 points)

Reflection (5 points)

1. Most challenging part of the workflow & why

The most challenging part of the workflow was the data strategy and preprocessing phase.

This stage required understanding what data sources the hospital would realistically collect, identifying missing or inconsistent data, ensuring privacy compliance, and designing appropriate feature engineering steps. Healthcare data is often sensitive, noisy, and unstructured, making it difficult to create a clean, fair, and representative dataset. Additionally, balancing ethical considerations while still building a useful model required thoughtful decision-making.

2. How I would improve my approach with more time/resources

With more time and resources, I would improve the project by:

- Collecting a larger and more diverse dataset to reduce bias and improve generalization.
- Using automated feature engineering tools (e.g., FeatureTools) to extract richer clinical features.
- Collaborating with medical professionals to better understand domain-specific variables and ensure the model reflects real clinical needs.
- Testing multiple advanced models (XGBoost, LightGBM, calibrated models) and comparing performance.
- Conducting a deeper fairness analysis using tools like AIF360 to identify disparities and ensure equitable predictions.

Diagram (5 points)

AI Development Workflow