ERICK WAMBUGU
AI FOR SOFTWARE ENGINEERING
POWER LEARN PROJECT AFRICA
WEEK 5

PART 1: SHORT ANSWER QUESTIONS

1. Problem Definition (6 points)

- Problem: Predicting short-term weather variables (temperature, rainfall probability, relative humidity) for the next 24–72 hours at station/grid-cell level.

3 objectives

- Produce accurate point forecasts for temperature and humidity (hourly).
- Produce calibrated probabilistic forecasts for precipitation (chance of rain).
- Deliver forecasts with low latency so downstream systems (alerts, irrigation control) can act in near real-time.

2 stakeholders

- National meteorological service / forecasters (use forecasts for warnings).
- Farmers & agritech platforms (use forecasts for irrigation and planting decisions).

1 KPI

- 24-hour RMSE for temperature (°C) — single, interpretable numeric indicator of short-term accuracy.

2. Data Collection & Preprocessing (8 points)

2 data sources

- Numerical Weather Prediction (NWP) model outputs (e.g., GFS/ECMWF fields) — gridded forecasts and atmospheric variables.
- Local observations: ground station time-series (temperature, humidity, rainfall), and optionally radar or satellite-derived precipitation estimates.

1 potential bias

- Spatial coverage bias: ground stations are denser in urban/wealthy areas, so the model may learn patterns that under-perform in rural or under-sampled regions (systematically worse forecasts where no stations exist).

3 preprocessing steps

- Temporal alignment & resampling: resample all sources to a common hourly time-step; align NWP lead times with observation timestamps.
- Handle missing data: fill short gaps with interpolation (linear/time-series methods), mark long gaps and drop or impute using model-based methods (e.g., seasonal decomposition or KNN).
- Feature scaling & augmentation: normalize continuous inputs (e.g., z-score per-station or per- grid) and add engineered features (hour-of-day, day-of-year, elevation, recent lags — t−1, t−3, t−6 hours).

ERICK WAMBUGU
AI FOR SOFTWARE ENGINEERING
POWER LEARN PROJECT AFRICA
WEEK 5

3. Model Development (8 points)

Model choice & justification

- Sequence model: Encoder–Decoder LSTM (or Temporal Convolutional Network). - Justification: weather forecasting is a time-series problem with temporal dependencies and multi-step outputs; LSTM/temporal convs capture temporal patterns, handle variable-length input windows, and can produce multi-horizon forecasts. If computation budget is limited, a Gradient Boosted Tree (XGBoost) on engineered lag features is a strong baseline.

Train/validation/test split

- Time-based split (no random shuffling):
- Train: earliest 70% of timeline,
- Validation: next 15% (for hyper-parameter tuning & early stopping),
- Test: final 15% (holdout period).
  This preserves temporal order and avoids leakage from future → past

2 hyper parameters to tune

- Learning rate — controls convergence speed and stability; crucial for avoiding under/overshooting and for generalization.
- Look-back window / sequence length (e.g., 24 vs 72 hours) — determines how much past context model uses; affects ability to learn diurnal and synoptic patterns.

4. Evaluation & Deployment (8 points)

2 evaluation metrics & relevance

- RMSE (Root Mean Squared Error) for continuous variables (temperature, humidity): penalizes large errors, widely interpretable in physical units (°C).
- Brier Score for binary/probabilistic events (rain/no-rain probability): measures calibration and accuracy of probability forecasts (lower is better)

What is concept drift? How to monitor it?

- **Definition:**
  - Concept drift occurs when the joint distribution of inputs and/or the relationship between inputs and targets changes over time (e.g., changing climate, new urban microclimate).
- **Monitoring approach:**
  - compute rolling performance metrics (RMSE, Brier) on recent data; compute statistical drift detectors (Population Stability Index or KL divergence on input features and residuals). Trigger alerts when metrics exceed thresholds and schedule model retraining or recalibration.

ERICK WAMBUGU
AI FOR SOFTWARE ENGINEERING
POWER LEARN PROJECT AFRICA
WEEK 5

1 technical deployment challenge

- Scalability & low-latency serving: serving high-resolution forecasts (many grid cells or stations) with multiple ensemble members can be computationally heavy. You must design efficient model serving (batching, model quantization, distributed inference, or using light surrogate models for fast updates) and ensure the pipeline can ingest real-time observations and NWP updates without becoming a bottleneck.

PART 2: CASE STUDY APPLICATION

Scenario: A hospital wants an AI system to predict patients re-admission risk within 30 days of discharge.

1. Problem Scope (5 points)

- **Problem:** Predict whether a discharged patient will be readmitted to the hospital within 30 days.

Objectives (3):

- Identify high-risk patients at discharge so care teams can arrange follow-up and reduce preventable re-admissions.
- Prioritize patients for post-discharge interventions (calls, home visits, medication reconciliation).
- Provide calibrated risk scores (probabilities) that clinicians can interpret and act on.

Stakeholders (at least 2):

- Clinicians / discharge planners (use predictions to decide interventions).
- Hospital administration / quality teams (reduce readmission penalties and improve care metrics).

2. Data Strategy (10 points)

Data sources

- Electronic Health Records (EHR): demographics, diagnoses (ICD codes), comorbidities, vitals, lab results, medications, previous admissions, discharge summaries.
- Operational & social data: prior utilization (ED visits), insurance status, scheduled follow-ups, social determinants (housing, living alone), and — where available — claims data.

Two ethical concerns

- Patient privacy / PHI risk: training on identifiable EHR data requires strict protections (access control, encryption, audit logging) and de-identification when possible. HIPAA provides de-identification methods and guidance.

ERICK WAMBUGU
AI FOR SOFTWARE ENGINEERING
POWER LEARN PROJECT AFRICA
WEEK 5

- Algorithmic bias & fairness: model may under perform for groups under-represented in training data (e.g., minorities, rural patients), leading to unequal resource allocation or harm; must assess fairness and adjust (reweighing, subgroup calibration).

Preprocessing pipeline (high-level)

1. Data extraction & harmonization

- Pull structured tables (demographics, encounters, meds, labs) and text (discharge note). Map to common schema; standardize code systems (ICD, LOINC, RxNorm).
- Timestamp alignment: ensure all features are relative to discharge time (lookback windows defined clearly).

2. De-identification & access control

- Remove direct identifiers for model development datasets or apply expert-determination de-identification; keep PHI only in secure, access-controlled environments for production linkage.

3. Missing data & cleaning

- Flag missingness (missingness itself can be predictive). Impute clinically where appropriate (last observation carried forward for recent vitals; model-based imputation for labs), but keep indicator flags.

4. Feature engineering

- Static features: age, sex, chronic conditions (e.g., Charlson score), social risk indicators.
- Temporal features: number of admissions in last 6/12 months, last vitals/lab trends (slopes), time since last discharge.
- Text features: extract key phrases from discharge summary (e.g., “left AMA”, “home oxygen”) via NLP; convert to bag-of-concepts or embeddings. (Hybrid structured + text models often yield gains).

5. Encoding & scaling

- Categorical encoding (target or one-hot), normalization of continuous features, creation of interaction features if clinically meaningful.

6. Splitting (temporal) & cohort definition

- Define index (discharge) and ensure no leakage (use time-based splits so training data precede validation/test). See Model Development below.

ERICK WAMBUGU
AI FOR SOFTWARE ENGINEERING
POWER LEARN PROJECT AFRICA
WEEK 5

3. Model Development (10 points)

Model selection & justification

- **Model:** Gradient Boosted Trees (e.g., XGBoost / LightGBM) or interpretable ensemble (with SHAP explanations); consider Logistic Regression baseline and a hybrid transformer/LSTM if adding unstructured text.
- **Why:** GBTs handle heterogeneous tabular EHR features, missing-ness, and nonlinearity well, are computationally efficient for training/serving, and feature importance/SHAP makes them interpretable for clinicians. More complex models (transformers on clinical text) can be added if text provides large gains.

Train/validation/test split (brief)

- **Time-based split to avoid temporal leakage**: train on earliest 70% of discharges, validate on next 15% (hyper parameter tuning + early stopping), final holdout test = last 15% (simulate future performance). This preserves temporal ordering and mimics deployment.

Confusion matrix (hypothetical) & precision/recall

Assume a test set of 1,000 discharged patients. You threshold the model so predicted readmit = positive.
Confusion matrix (counts):

	Predicted Positive	Predicted Negative	Row total
Actual Positive	TP = 120	FN = 80	200
Actual Negative	FP = 30	TN = 770	800
Column total	150	850	1000

(Checks: 120+80 = 200 actual positives; 30+770 = 800 actual negatives; totals = 1000.)

- **Precision** = TP / (TP + FP) = 120 / (120 + 30) = 120 / 150 = 0.80 (80%).
- **Recall (Sensitivity)** = TP / (TP + FN) = 120 / (120 + 80) = 120 / 200 = 0.60 (60%).
- **F1 score** = 2 * (Precision * Recall) / (Precision + Recall) = 2*(0.8*0.6)/(1.4) = 0.6857 ≈ 0.69.

- **Interpretation:** Model is precise (when it flags readmission risk it's likely correct) but misses 40% of true re admissions (recall 60%) — may be acceptable if resource constraints mean prioritizing high-certainty interventions, but consider adjusting threshold to improve recall if care priority is to catch more at-risk patients.

ERICK WAMBUGU
AI FOR SOFTWARE ENGINEERING
POWER LEARN PROJECT AFRICA
WEEK 5

4. Deployment (10 points)

Steps to integrate the model into the hospital's system

1. Proof of concept & clinician validation

- Run model silently (no clinician-facing alerts) for a pilot period; present predictions retrospectively to clinician panels to assess clinical utility and workflow fit. (Clinical evaluation & human-in-the-loop validation).

2. Technical integration

- Wrap model as an API/service (containerized, e.g., Docker) that can accept a discharge event and return risk score and top contributing features (explanations).
- Integrate with EHR via standard interfaces (HL7 FHIR where possible) so that when a patient is discharged, the EHR calls the model or the model subscribes to discharge events. Use robust message queueing for reliability.

3. UI/Workflow

- Embed risk score into clinician dashboard or discharge planner view (the "right person, right information, right time" concept). Provide actionable recommendations and explainability (top features or SHAP).

4. Monitoring & logging

- Log predictions, model inputs (hashed/anonymized as needed), and outcomes for continual evaluation. Implement performance dashboards for key metrics and data drift detectors.

5. Pilot & phased rollout

- Start in a single department/unit, collect clinician feedback, measure impact (e.g., readmission reduction, false alert burden), then scale.

6. Governance

- Establish multidisciplinary oversight: clinicians, data scientists, IT, compliance/legal, and a process for updates, incident response, and model retirement.

ERICK WAMBUGU
AI FOR SOFTWARE ENGINEERING
POWER LEARN PROJECT AFRICA
WEEK 5

Ensuring HIPAA & regulatory compliance

- Data minimization & de-identification: Use de-identified data for model development when possible; if PHI is needed, keep it within HIPAA-covered environments and apply expert-determination de-identification methods where appropriate.
- Access control & encryption: Encrypt PHI at rest and in transit, enforce least privilege access, multi-factor auth, and detailed audit logs for who accessed data and when.
- Business Associate Agreements (BAAs): Ensure any cloud/third-party service handling PHI signs BAAs and follows HIPAA safeguards.
- Validation & documentation: Maintain model development records, validation reports, and clinical risk assessments (for accountability and potential regulatory review). Establish process for incident reporting and periodic privacy/security audits.

5. Optimization (5 points)

One method to address over-fitting:

- Use regularization + early stopping with cross-validation. For tree-based models this includes limiting tree depth, increasing minimum child weight or leaf size, and applying L1/L2 regularization (or using subsampling/column subsampling). Combine with early stopping on the temporal validation set to prevent overfitting to historical idiosyncrasies. Additionally, use temporal cross-validation (rolling windows) to ensure generalization across time.

PART 3: CRITICAL THINKING

1. Ethics & Bias (10 points)

How biased training data might affect patient outcomes

- If the training data reflects historical biases — for example, under representation of certain groups (elderly, low-income, rural, or minority patients) — the model may systematically under predict readmission risk for these populations.
- **Consequences include:**
  - **Unequal care:** vulnerable patients may not be flagged for follow-up, leading to higher real readmission rates.
  - **Worsening health disparities:** reinforces existing inequities in healthcare delivery.
  - **Loss of trust:** patients and clinicians lose confidence in AI tools if they observe biased recommendations.

ERICK WAMBUGU
AI FOR SOFTWARE ENGINEERING
POWER LEARN PROJECT AFRICA
WEEK 5

One strategy to mitigate bias

Bias auditing and re balancing:

- Before training, analyze performance across subgroups (age, gender, socioeconomic, ethnicity) to identify disparities.
- Apply reweighing or oversampling techniques (e.g., SMOTE for minority groups) to balance representation.
- During evaluation, use fairness metrics (e.g., equal opportunity, disparate impact ratio).
- Combine with clinician review — ensure decisions are medically sound, not only statistically fair.

PART 4: REFLECTION AND WORKFLOW DIAGRAMS

Most challenging part of the workflow

- The **data preprocessing and feature engineering** phase was the most challenging.
  EHR data are messy, incomplete, and inconsistent across departments. Handling missing lab values, merging structured and unstructured records (clinical notes), and ensuring temporal consistency required significant time and domain understanding.
  Moreover, balancing privacy (de-identification) with data richness added another layer of complexity.

How to improve with more time/resources

With additional resources:

1. **Invest in better data infrastructure —** a unified clinical data warehouse to streamline extraction and cleaning.
2. **Collaborate with clinical experts —** to co-design features that capture real-world clinical reasoning.
3. **Implement automated ML pipelines —** for continuous model retraining and bias monitoring.
4. **Use advanced NLP models —** to leverage insights from discharge summaries and physicians' notes.

ERICK WAMBUGU
AI FOR SOFTWARE ENGINEERING
POWER LEARN PROJECT AFRICA
WEEK 5

Below is a **flowchart of the AI Development Workflow** for the hospital's readmission prediction system:

```
1. Problem Definition
Define goal: predict 30-day
readmission risk
```

```
2. Data Collection
Gather EHR, demographics,
lab results, notes
```

```
3. Data Preprocessing
Clean, handle missing data,
feature engineering,
de-identify patient info
```

```
4. Model Development
Choose model (e.g. XGBoost)
Train, tune, validate
```

```
5. Evaluation
Use metrics: Accuracy,
Precision, Recall, AUC
```

```
6. Deployment
Integrate with hospital EHR
Ensure HIPAA compliance
```

```
7. Monitoring & Feedback
Track performance, detect
concept drift, retrain
```

ERICK WAMBUGU
AI FOR SOFTWARE ENGINEERING
POWER LEARN PROJECT AFRICA
WEEK 5

Workflow summary:
Each stage feeds into the next — from defining the problem to continuous monitoring after deployment. This iterative cycle ensures the model stays accurate, ethical, and aligned with clinical needs.