Part 1: Short Answer Questions (30 points)

Hypothetical AI Problem

A healthcare clinic wants to develop an **AI-based predictive system** that identifies patients who are **likely to miss their upcoming medical appointments**. By anticipating no-shows, the clinic can take preventive actions such as sending reminders, rescheduling, or offering telehealth options.

Objectives (3)

1. **Predict no-shows accurately** using patient and appointment data.
2. **Reduce missed appointments** by enabling targeted communication with at-risk patients.
3. **Optimize clinic operations** through improved scheduling and reduced idle doctor time.

Stakeholders (2)

1. **Clinic administrators** – use predictions to adjust schedules and resource allocation.
2. **Healthcare providers (doctors/nurses)** – rely on accurate scheduling to manage workload efficiently.

Key Performance Indicator (KPI)

- **Appointment Attendance Accuracy (Model Accuracy or F1-Score):** Measures how well the AI model correctly predicts actual attendance or no-shows.
  (Higher accuracy/F1 = more reliable decision support for clinic operations.)

2. Data Collection & Preprocessing (8 points)

Data Sources (2)

1. **Electronic Health Records (EHR):** Patient demographics, medical history, and past appointments.

2. **Appointment Scheduling System:** Details about appointment date, reminder messages, and previous attendance behavior.

Potential Bias

- **Socioeconomic or location bias:**
  Patients from rural or low-income areas may have higher no-show rates due to transportation or cost issues.
  If not balanced, the model might unfairly predict higher no-show probabilities for these groups, leading to biased interventions.

Preprocessing Steps (3)

1. **Handle Missing Data:** Replace missing demographic or reminder-sent values using imputation.
2. **Encode Categorical Variables:** Convert gender, reminder status, and other categorical fields into numerical form using one-hot or label encoding.
3. **Normalize Numerical Features:** Scale features such as *distance from clinic* and *age* to ensure fair model weighting.

---

3. Model Development (8 points)

Chosen Model & Justification

- **Model: Random Forest Classifier**
- **Justification:**
  - Handles both numerical and categorical data efficiently.
  - Resistant to overfitting and provides feature importance insights.
  - Performs well on medium-sized structured datasets common in healthcare records.

Data Splitting

- **Training Set (70%)** – Used to train the model.
- **Validation Set (15%)** – Used for hyperparameter tuning and model selection.
- **Test Set (15%)** – Used to evaluate final model performance on unseen data.

Hyperparameters to Tune (2)

1. **Number of Trees (n_estimators):** Controls how many decision trees are built. More trees can improve accuracy but increase computation time.
2. **Maximum Depth (max_depth):** Limits the tree depth to prevent overfitting and improve generalization.

---

4. Evaluation & Deployment (8 points)

Evaluation Metrics (2)

1. **Precision:** Measures the proportion of patients predicted as no-shows who actually missed their appointments. Helps reduce false alarms.
2. **Recall (Sensitivity):** Measures how many actual no-shows were correctly identified. Important for ensuring high-risk patients are not missed.

(Together, Precision and Recall capture the balance between over- and under-predicting no-shows.)

Concept Drift

- **Definition:**
  Concept drift occurs when **the relationship between input features and target outcomes changes over time**, causing model performance to degrade.
  (Example: After introducing a new SMS reminder system, fewer patients miss appointments, changing prediction patterns.)
- **Monitoring Approach:**
  Continuously track model accuracy and F1-score on new data. Retrain the model periodically using the latest appointment data to adapt to changes.

Technical Challenge During Deployment

- **Scalability:**
  As appointment volume grows, the model's real-time prediction API (e.g., built in Flask or deployed on Azure) must handle increasing request loads efficiently.
  Solution: Use cloud-based deployment (Azure ML or Kubernetes) with autoscaling to maintain response speed.

Part 2: Case Study Application (40 points)

Scenario: A hospital wants an AI system to predict patient readmission risk within 30 days of discharge.

Tasks:

1. **Problem Scope (5 points):** Define the problem, objectives, and stakeholders.
2. **Data Strategy (10 points):**
   - Propose data sources (e.g., EHRs, demographics).
   - Identify **2 ethical concerns** (e.g., patient privacy).
   - Design a preprocessing pipeline (include feature engineering steps).
3. **Model Development (10 points):**
   - Select a model and justify it.
   - Create a confusion matrix and calculate precision/recall (hypothetical data).
4. **Deployment (10 points):**
   - Outline steps to integrate the model into the hospital's system.
   - How would you ensure compliance with healthcare regulations (e.g., HIPAA)?
5. **Optimization (5 points):** Propose **1 method** to address overfitting.

1. Problem Scope (5 points)

Problem Definition

The clinic faces frequent **appointment no-shows**, leading to wasted medical staff time, scheduling inefficiencies, and reduced patient care quality. The goal is to build an **AI model that predicts which patients are likely to miss upcoming appointments**, allowing proactive intervention.

Objectives

1. **Predict appointment attendance** using historical and behavioral data.
2. **Reduce no-show rates** through targeted reminders and outreach.
3. **Improve operational efficiency** by optimizing scheduling and resource allocation.

Stakeholders

- **Clinic Administrators:** Use predictions to reschedule or double-book efficiently.
- **Healthcare Providers:** Optimize workload planning and reduce idle time.
- **Patients:** Benefit from improved reminder systems and personalized communication.

2. Data Strategy (10 points)

Proposed Data Sources

1. **Electronic Health Records (EHR):**
   - Patient demographics, medical history, chronic conditions, and previous attendance.
2. **Appointment Scheduling System:**
   - Date/time, reminder sent (SMS, call, email), day of week, waiting time, and appointment type.

Ethical Concerns (2)

1. **Patient Privacy:**
   - Personal health information (PHI) must be securely stored and processed to comply with privacy regulations (e.g., HIPAA).
2. **Algorithmic Bias:**
   - Model might unfairly classify certain demographic groups (e.g., older or rural patients) as high-risk if data is imbalanced.

Preprocessing Pipeline (with Feature Engineering)

Step	Description
1. Data Cleaning	Handle missing demographic or reminder data using imputation (mean/mode).
2. Encoding Categorical Features	Convert gender, day of week, and appointment type into numeric form using one-hot encoding.
3. Normalization	Scale continuous variables (e.g., distance, age, waiting time) using StandardScaler.
4. Feature Engineering	Create new features such as: - time_gap = days since last visit - reminder_effectiveness = ratio of attended appointments after reminders - appointment_hour (morning vs. afternoon).
5. Train/Test Split	Split data (70% training, 15% validation, 15% test).

---

3. Model Development (10 points)

Selected Model

Model: Random Forest Classifier
Justification:

- Handles nonlinear relationships between features and target variable.
- Works well on tabular healthcare data with mixed types (categorical + numerical).
- Provides feature importance for interpretability, critical in healthcare decisions.

Hypothetical Confusion Matrix

	Predicted: No-Show	**Predicted: Show**
Actual: No-Show	40	10
Actual: Show	15	85

Precision and Recall

Precision=TPTP+FP=4040+15=0.727 (72.7%)\text{Precision} = \frac{TP}{TP + FP} = \frac{40}{40 + 15} = 0.727 \ (72.7\%)Precision=TP+FPTP=40+1540 =0.727 (72.7%) Recall=TPTP+FN=4040+10=0.8 (80%)\text{Recall} = \frac{TP}{TP + FN} = \frac{40}{40 + 10} = 0.8 \ (80\%)Recall=TP+FNTP =40+1040=0.8 (80%)

Interpretation:
The model correctly identifies 80% of patients who will miss appointments (high recall) and is right about 73% of its no-show predictions (moderate precision). This is acceptable since missing a real no-show is costlier than issuing extra reminders.

---

4. Deployment (10 points)

Integration Steps

1. **Model Packaging:**
   Save the trained model and preprocessing pipeline as .pkl files using pickle or joblib.
2. **Create REST API (Flask or FastAPI):**
   Develop an endpoint /predict that accepts patient data and returns the no-show probability.
3. **System Integration:**
   - Connect the API with the hospital's scheduling software (e.g., via secure HTTPS).
   - Automatically trigger reminders for high-risk patients.
4. **Monitoring:**
   Track prediction accuracy and update the model periodically using recent appointment data.
5. **Visualization Dashboard (Optional):**
   Use Dash or Streamlit to display trends and performance metrics to administrators.

Compliance with Healthcare Regulations (HIPAA)

To ensure regulatory compliance:

- **Data Encryption:** Use HTTPS and encrypt PHI both at rest and in transit.
- **Access Control:** Restrict data access to authorized staff only.
- **De-identification:** Remove personally identifiable information during training.
- **Audit Logging:** Maintain logs of all model predictions and data access activities.

---

5. Optimization (5 points)

Method to Address Overfitting

Technique: Cross-Validation and Regularization

- Use **k-fold cross-validation** to validate performance across multiple data subsets, reducing variance.
- In Random Forest, limit **max_depth** and **min_samples_split** to prevent overly complex trees that memorize training data.

Part 3: Critical Thinking (20 points)

1. **Ethics & Bias (10 points):**
    - How might biased training data affect patient outcomes in the case study?
    - Suggest **1 strategy** to mitigate this bias.
2. **Trade-offs (10 points):**
    - Discuss the trade-off between model interpretability and accuracy in healthcare.
    - If the hospital has limited computational resources, how might this impact model choice?

1. How Biased Training Data Might Affect Patient Outcomes

Biased training data can lead to **unfair or harmful predictions** that disproportionately affect certain patient groups. Examples include:

- **Demographic Bias:**
  If historical data shows that patients from low-income or rural areas miss more appointments (possibly due to transportation or communication barriers), the model might **overpredict no-shows** for these groups.
  → This could result in them receiving fewer appointment slots or less follow-up care — **worsening healthcare inequality**.
- **Data Representation Bias:**
  If younger patients or those using digital reminders are overrepresented in the dataset, the model may perform poorly for elderly patients or those without smartphone access.
  → This can **skew predictions** and reduce trust in AI-assisted scheduling.
- **Outcome Bias:**
  The model might reinforce historical inefficiencies — if the clinic previously under-served certain communities, the AI could continue to do so unintentionally.

---

2. Strategy to Mitigate Bias

Strategy: Use **Fairness-Aware Reweighting with IBM AI Fairness 360 (AIF360)**

- **Step 1:** Detect bias using metrics like *disparate impact* or *equal opportunity difference*.
- **Step 2:** Apply **reweighting** — assign higher weights to underrepresented groups so the model learns balanced decision boundaries.
- **Step 3:** Reassess model performance using fairness metrics alongside accuracy (e.g., demographic parity).
- **Step 4:** Combine with **human oversight** — clinicians should review flagged predictions for fairness and plausibility.

✅ **Outcome:** This ensures the model's recommendations are equitable across age, gender, and socioeconomic groups, promoting fairness in patient outreach and treatment.

---

7. Trade-offs (10 points)

1. Interpretability vs. Accuracy

Aspect	High Interpretability (e.g., Logistic Regression, Decision Tree)	High Accuracy (e.g., Random Forest, Neural Network)
Transparency	Easy to explain to doctors and regulators	Often a "black box" — harder to justify decisions
Trust in Healthcare	Higher — clinicians understand reasoning	Lower — lack of explainability may hinder adoption
Performance	May miss subtle patterns in complex data	Captures nonlinear relationships and improves accuracy
Use Case Fit	Preferred when decisions affect patient safety or require justification	Useful for large datasets where predictive performance is key

🔹 **In healthcare**, interpretability is often **prioritized over raw accuracy**, because medical staff and regulators must understand *why* the model flagged a patient as a likely no-show.
However, a **balanced approach** can be achieved using **explainable AI tools** like *SHAP (SHapley Additive exPlanations)* to interpret complex models.

---

2. Impact of Limited Computational Resources on Model Choice

If the hospital has **limited computing infrastructure**, it directly influences which AI models can be deployed:

- **Constraints:**
    - Limited CPU/GPU power may make deep learning or ensemble methods (e.g., Gradient Boosting) impractical.

- Training times may be long and predictions slow.

- **Practical Model Choice:**
  - Prefer **lightweight, interpretable models** such as **Logistic Regression**, **Decision Tree**, or **Naïve Bayes**.
  - These models:
    - Train quickly on smaller datasets
    - Require less memory and processing power
    - Are easier to deploy in low-resource environments (e.g., local servers)

✅ **Trade-off:** Slightly lower predictive accuracy but **faster, cheaper, and more transparent**, which aligns better with healthcare's ethical and operational constraints.

Part 4: Reflection & Workflow Diagram (10 points)

1. **Reflection (5 points):**
   - What was the most challenging part of the workflow? Why?
   - How would you improve your approach with more time/resources?
2. **Diagram (5 points):**
   - Sketch a flowchart of the AI Development Workflow, labeling all stages.

Reflection (5 points)

1. Most Challenging Part of the Workflow

The most challenging part of the project was **handling data quality and bias during preprocessing**.

- Patient data from multiple systems (EHR, scheduling, and reminders) often contain **missing values**, inconsistent entries, or duplicated records.
- Additionally, ensuring the dataset was **fair and representative** was difficult because some demographic groups (e.g., elderly or rural patients) had fewer records, increasing the risk of biased predictions.

This stage required balancing **data cleaning**, **ethical fairness**, and **model performance**, which is complex in healthcare applications.

---

2. How I Would Improve the Approach with More Time and Resources

If given more time and resources, I would:

1. **Collect richer contextual data** – e.g., transportation availability, socioeconomic factors, and communication preferences to improve prediction accuracy.
2. **Integrate Explainable AI (XAI) tools** such as **SHAP or LIME** to make model predictions more transparent and trustworthy for clinicians.
3. **Develop a live monitoring dashboard** to track real-time model performance, concept drift, and fairness metrics over time.
4. **Collaborate with medical staff** for continuous feedback — ensuring the system aligns with clinical judgment and ethical standards.

✅ These improvements would lead to a **more reliable, fair, and deployable** AI system suitable for real-world healthcare environments.

Diagram (5 points)

Below is a **flowchart** of the complete **AI Development Workflow** for the clinic appointment no-show prediction model:

AI Development Workflow Diagram

```
┌──────────────────────────────────────┐
│   Problem Definition                 │
│ - Identify objectives                │
│ - Define stakeholders                │
└──────────────────────────────────────┘
        │
        ▼
┌──────────────────────────────────────┐
│   Data Collection                    │
│ - Gather EHR + scheduling            │
│   system data                        │
└──────────────────────────────────────┘
        │
        ▼
┌──────────────────────────────────────┐
│ Data Preprocessing                   │
│ - Clean, encode, normalize           │
│ - Feature engineering                │
│ - Handle bias & missing              │
└──────────────────────────────────────┘
        │
```

▼

Model Development
- Select algorithm
- Tune hyperparameters
- Split data (train/test)

▼

Model Evaluation
- Confusion matrix
- Precision, Recall, AUC
- Fairness assessment

▼

Deployment
- Integrate with clinic system (API)
- Monitor drift & KPIs

▼

Continuous Improvement
- Retrain with new data
- Address drift & bias
