AI Development Workflow Assignment

Part 1: Short Answer Questions

1. Problem Definition
- Hypothetical Problem: Predicting breast cancer malignancy from diagnostic features (e.g., tumour size, cell morphology).
- Objectives:
    1. Achieve ≥90% F1-score in malignancy classification.
    2. Reduce false negatives to <5% (critical to avoid missed diagnoses).
    3. Provide interpretable predictions for clinician trust.
- Stakeholders: Patients, Oncologists.
- KPI: F1-score (balances precision/recall, crucial for medical decisions).

---

2. Data Collection & Preprocessing
- Data Sources:
    1. Kaggle breast cancer dataset (features like tumor_radius, texture).
    2. Hospital biopsy reports (augmenting with patient history).
- Potential Bias: Underrepresentation of rare subtypes (e.g., triple-negative breast cancer in minority populations), leading to poor generalisation.
- Preprocessing Steps:
    1. Impute missing values (e.g., median imputation for mitosis counts).
    2. Normalise features (e.g., scale cell_area to [0,1] via Min-Max).
    3. Balance classes (SMOTE oversampling for minority malignant cases).

---

3. Model Development
- Model Choice: Random Forest (justification: handles high-dimensional data, robust to outliers, provides feature importance for clinical interpretability).
- Data Splitting: 70% training, 15% validation (hyperparameter tuning), 15% test (final evaluation). Stratified sampling to preserve class ratios.
- Hyperparameters:
    1. n_estimators (optimise tree count to balance accuracy/compute time).
    2. max_depth (prevent overfitting by limiting tree complexity).

---

4. Evaluation & Deployment
- Evaluation Metrics:
    1. F1-score: Prioritises both false positives (unnecessary stress) and false negatives (missed cancer).
    2. AUC-ROC: Measures the separability of malignant/benign classes across thresholds.
- Concept Drift: When data distributions shift post-deployment (e.g., new imaging technology). Monitoring: Track F1-score weekly; trigger retraining if performance drops >5%.

- Technical Challenge: Scalability for real-time predictions. Solution: Containerize model with Docker; deploy on cloud GPUs.

Part 2: Case Study Application

Problem Scope

- Problem: Predict 30-day hospital readmission risk.
- Objectives:
    1. Identify high-risk patients for proactive care.
    2. Reduce readmissions by ≥20%.
    3. Minimise false negatives (high-risk patients missed).
- Stakeholders: Patients, Clinicians, Hospital Administrators.

Data Strategy

- Data Sources:
    1. EHRs (lab results, medications).
    2. Socioeconomic data (e.g., ZIP code → access to care).
- Ethical Concerns:
    1. Privacy: Anonymise data (HIPAA compliance).
    2. Bias: Overrepresentation of affluent patients → underdiagnosis in low-income groups.
- Preprocessing Pipeline:
    1. Clean missing lab values (KNN imputation).
    2. Encode categorical variables (e.g., diagnosis_code → one-hot).
    3. Feature Engineering:
        - Create comorbidity_score (sum of chronic conditions).
        - Calculate medication_adherence (prescriptions vs. refills).

Model Development

- Model Choice: Random Forest (handles mixed data types; explains risk factors via feature importance).
- Confusion Matrix (Hypothetical 1,000 patients):

	Predicted: No	Predicted: Yes
Actual: No	700 (TN)	50 (FP)
Actual: Yes	80 (FN)	170 (TP)

    - Precision = TP/(TP+FP) = 170/(170+50) = 77.3%
    - Recall = TP/(TP+FN) = 170/(170+80) = 68.0%

Deployment

- Integration Steps:
    - Build REST API with Flask.

    - Integrate with hospital EHR via FHIR standards.
    - Output risk scores to clinician dashboards.
- Compliance:
    - HIPAA: Encrypt data in transit (TLS) and at rest (AES-256); audit logs for data access.
    - GDPR: Patient consent workflows for data usage.

Optimization

- Overfitting Solution: Feature importance pruning (remove low-impact features like redundant lab tests) to simplify the model.

Part 3: Critical Thinking

Ethics & Bias

- Bias Impact: Overrepresentation of urban populations could mispredict rural patient risk (e.g., distance to clinics not captured), leading to inadequate care.
- Mitigation: Stratified sampling by rural/urban residency during training; fairness-aware loss functions.

Trade-offs

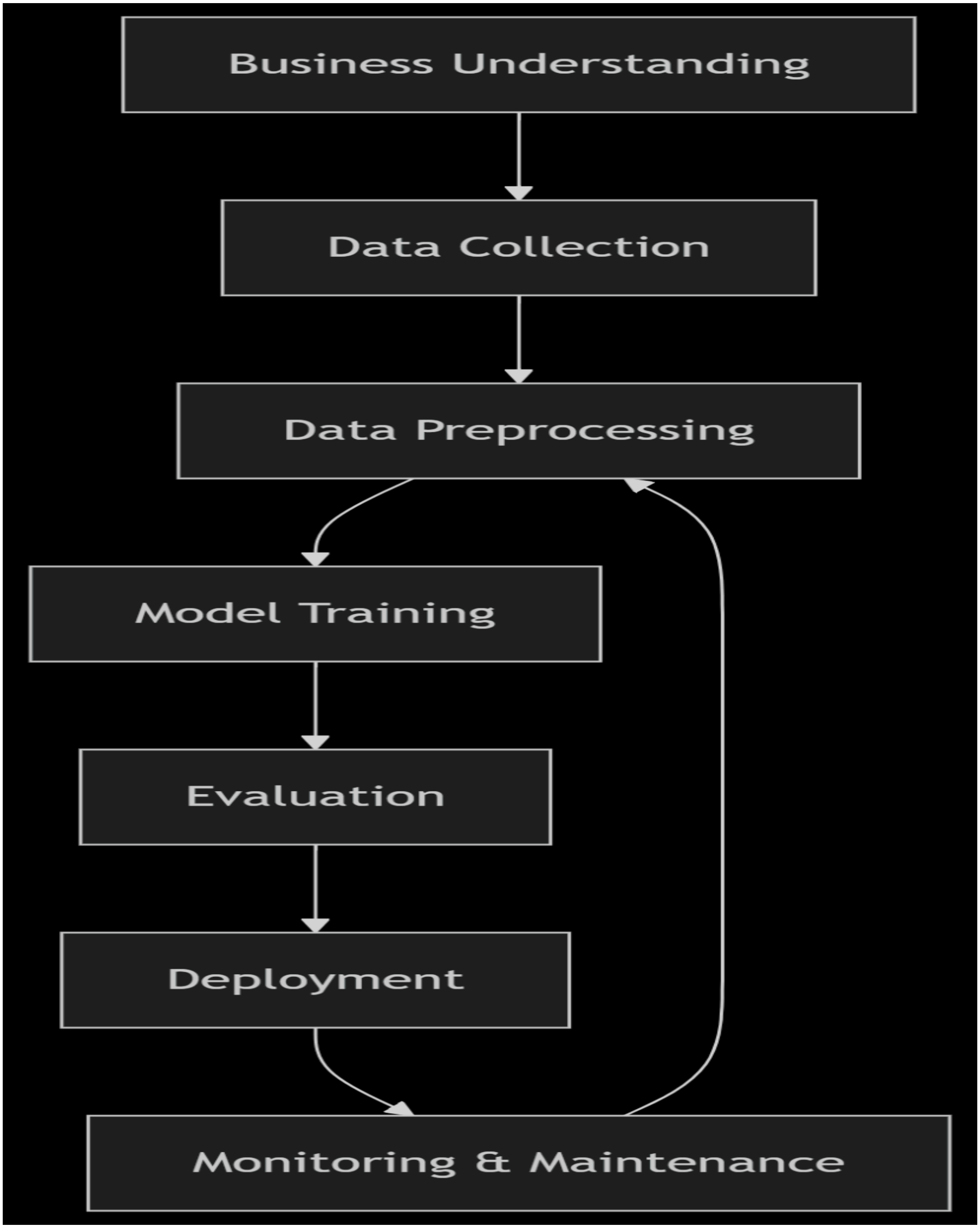
- Interpretability vs. Accuracy:
    - Random Forest (interpretable) may have lower accuracy than neural networks (black-box).
    - Resolution: Use SHAP values to explain complex models → maintain both accuracy and trust.
- Limited Resources: Opt for logistic regression (faster training) over ensemble methods; prioritise critical features to reduce dimensionality.

Part 4: Reflection & Workflow Diagram

Reflection

- Biggest Challenge: Bias mitigation (e.g., ensuring underrepresented groups are modeled fairly). Requires domain expertise to identify sensitive variables.
- Improvements:
    1. Partner with diverse hospitals to expand data coverage.
    2. Implement continuous bias monitoring with IBM AIF360.

Workflow Diagram:



Key Stages:

1. Business Understanding: Define objectives (e.g., "reduce readmissions").
2. Data Collection: EHRs, demographics.
3. Preprocessing: Imputation, normalisation, feature engineering.
4. Model Training: Random Forest with hyperparameter tuning.
5. Evaluation: F1-score, AUC-ROC, fairness metrics.
6. Deployment: API integration with EHRs.
7. Monitoring: Track drift, bias, and performance.

Sources:

- CRISP-DM framework.
- HIPAA compliance guidelines.
- IBM AIF360 documentation.