Theoretical Foundations Report

Theme: Building Intelligent Software Solutions

1. Problem Definition

In this hypothetical AI project, the goal is to **predict whether a patient will be readmitted to a hospital within 30 days of discharge**. Hospital readmissions are costly and often preventable, so an AI model could help medical professionals take proactive steps to reduce risk.

Objectives:

1. Build an AI model that predicts readmission risk with high accuracy.
2. Identify key risk factors (e.g., diagnosis, age, discharge condition).
3. Provide decision support for doctors to prioritize patient follow-up care.

Stakeholders:

- Hospital administrators (cost control)
- Healthcare providers (treatment planning)

Key Performance Indicator (KPI):

- **F1-score** of the model on test data, ensuring a balance between precision and recall for detecting high-risk patients.

---

2. Data Collection & Pre-processing

To build this model, we would need access to diverse and relevant data.

Data Sources:

1. Electronic Health Records (EHR) including clinical notes, discharge summaries, and diagnosis codes.
2. Patient demographic and insurance records: age, gender, socio-economic status.

Potential Bias:
A major risk is **underrepresentation of certain groups**, like rural or low-income patients. This can cause the model to generalize poorly and lead to unequal care.

Pre-processing Steps:

1. **Handling missing values** – using imputation (mean/mode) or removing rows with too much missing data.
2. **Normalization** – scaling numerical features like age or length of stay to prevent bias during model training.
3. **Encoding** – converting categorical variables (e.g., diagnosis codes, gender) into numerical format using one-hot encoding.

---

3. Model Development

Given the tabular nature of the healthcare data, a **Random Forest classifier** is appropriate. It can handle mixed data types, is resistant to overfitting, and gives feature importance insights.

Data Splitting Strategy:

- **Training set:** 70%
- **Validation set:** 15%
- **Test set:** 15%

  This ensures we have enough data to train, tune, and evaluate fairly.

Hyper-parameters to Tune:

1. n_estimators (number of trees): Higher numbers improve performance but increase training time.
2. max_depth: Controls how deep trees grow, helping avoid overfitting on small datasets.

---

4. Evaluation & Deployment

To measure model performance, we'll use:

1. **Accuracy** – to measure the overall correctness of the predictions.
2. **F1-score** – to balance precision and recall, especially useful when readmission cases are rare (imbalanced classes).

Concept Drift:
Concept drift refers to changes in the data patterns over time (e.g., new illnesses, policy changes). We can monitor this by:

- Tracking drops in model accuracy over time
- Setting up scheduled retraining using new hospital data

- **Deployment Challenge:**

Integrating the AI model into hospital IT systems may require compatibility with legacy infrastructure and strict data privacy policies (like HIPAA compliance), which can delay deployment or require custom solutions.