# Part 1

## Hypothetical AI Problem:

Predicting high-revenue Uber rides to optimize fleet allocation and pricing.

## Objectives:

1. Identify ride patterns that yield above-average revenue per mile.
2. Reduce cancellation rates in high-demand zones.
3. Recommend fleet mix adjustments to increase average revenue per ride.

## Stakeholders:

1. Operations Manager (responsible for fleet deployment)
2. Revenue Analyst (monitors pricing and profitability)

## Key Performance Indicator (KPI):

Average Revenue per Ride (calculated from completed rides only).

## Data Collection & Preprocessing

## Data Sources:

1. Uber ride logs- includes fare, distance, vehicle type, status, rating, and payment method.
2. City zoning data - maps ride coordinates to urban, suburban, or rural zones.

## Potential Bias:

Payment method bias - cash-heavy regions may show higher cancellations, not due to rider intent but due to driver preferences or app limitations.

## Preprocessing Steps:

1. Handle missing data -impute missing ratings or flag them for modeling.
2. Normalize fare and distance - scale values for fair comparison across cities.
3. Filter anomalies - exclude rides with zero distance and non-zero fare from revenue calculations.

## Model Development

**Chosen Model:**

Random Forest Classifier - robust to outliers, handles mixed data types, and provides feature importance for stakeholder insights.

Data Splitting:

- Training Set: 70%
- Validation Set: 15%
- Test Set: 15%
- Stratified by city and vehicle type to preserve distribution.

**Hyperparameters to Tune:**

1. max_depth - controls model complexity and overfitting.
2. n_estimators - number of trees; affects performance and stability.

**Evaluation & Deployment**

**Evaluation Metrics:**

1. Precision (High Revenue Segment)- ensures recommendations don't misclassify low-revenue rides as high-value.
2. Recall (High Revenue Segment) -captures most of the truly profitable rides for strategic action.

**Concept Drift:**

Definition: When the relationship between input features and target variable changes over time (e.g. rider behavior shifts due to economic factors).

Monitoring Strategy:

- Track KPI trends monthly.
- Revalidate model predictions against new ride data.
- Use drift detection algorithms (e.g., Kolmogorov-Smirnov test).

**Deployment Challenge:**

Scalability- real-time prediction across thousands of rides per minute requires optimized infrastructure and parallel processing.

## Part 2

**Part 2: Case study application**

**The problem**

There are a number of problems or risks associated patient readmission (within 30 days)

- **Time**

Time is often limited when patients are being discharged, which means staff may not have enough opportunity to give clear instructions, review medications, or make sure patients fully understand warning signs to watch for. This can put both the patient and the wider community at risk. When health care workers don't have enough time, the quality of their work can decline, and this ultimately leads to poorer conditions within the hospital.

- **Financial Constraints**

Financial constraints also play a major role in increasing the risk of readmissions. When hospitals face limited budgets, they may not have enough staff, resources, or time to properly support each patient during discharge. This can mean fewer nurses available, rushed assessments, or reduced access to follow-up services. Patients with financial difficulties may struggle to afford medications, transportation to appointments, or home care support. All of these gaps increase the chances of complications after discharge, putting the patient's health—and the overall health system—under even more strain.

- **Exposure to healthcare associated bugs**

Exposure to healthcare-associated bugs is another serious factor that increases the risk of patient readmission. Hospitals are busy environments where many people with different illnesses are treated in close proximity. When patients spend more time in the hospital—especially if their stay is prolonged or their immunity is low—they are more likely to encounter bacteria and other pathogens that circulate within the facility. If proper infection-control practices are rushed or compromised due to limited staff or time, patients may be exposed to these harmful bugs. This

can lead to new infections after they go home, forcing them to return for further treatment. As a result, hospital-acquired infections not only threaten the wellbeing of patients but also place additional pressure on the healthcare system through preventable readmissions.

**Objectives**

Identify how many admissions are there in one month

Identify how many readmissions are there in a month

What are the trends of readmission in 30 days-Identify the average of all admissions per month

**Stakeholders**

- Patients (general public)
- Health care workers

**Tasks: Data strategy**

Hospital database which is gotten directly from patients when they visit healthcare centres. It can be Electrical database, medical aid database. The emphasis is on the importance of data collection and accuracy.

This database can be divided into the following:

Demographics-these are groups of patients who visit the hospital. For instance a group of 14 year olds being admitted at the same time, turned out they are all from a same place and their stomach problems happened to be due to malaria caused by contaminated water at their school.

Post admission data-usually collected on readmission- too often times hospitals are not concerned or rather too focused on what happens to the patient after being discharged, unless they are readmitted then they are asked a series of questions to identify any factors causing illnesses or find out if other people will flood the hospital in no time.

Lifestyle-there might be lifestyle choices that cause certain illnesses or symptoms in patients for example smokers may show high levels of blood pressure while heavy drinkers may end up

experiencing liver failure in the long run. Same as unhealthy living that causes illnesses like stroke, hypertension and so forth.

**Ethical concerns**

- Biases and fairness

This refers to unbalanced data that causes misinterpretation of data. Healthcare workers may apply biases instead of asking questions to a patient so the get inaccurate results, therefore affect the remedies to health problems.

- Confidentiality

Patients should be free in sharing information about them knowing full well that their information will not end up in the wrong hands and that everyone who has access to private information will use it responsibly i.e. keeping in mind the ethics that govern the health care systems which are: autonomy, benefice, non-maleficence, justice.

**Design a pre-processing pipeline**

The first step in pre-processing is data collection. In this stage we will be pulling out data from hospital database.

The second stage is to cleanse data. Remember the data we will use is not only our data, we are to remove data we do not need and focus on the most accurate relevant data. We organise data such that it does not have duplicates, fill in missing figures and ensure data is standardised.

The third step is to transform data in this stage we check if there is any continuous data and categorise it.

Next step we focus on the identified objectives i.e. identify the exact number the patient has been admitted in the hospital (Previous admissions) for the past 30 days. Create the average of a patient's visit for 30 days. Identify their length of stay and compare their discharge date-to their admission date. Creating a gap in days between discharged date to readmission date.

Features selection is where we pay attention to contributing factors to their hospitalization e.g. were they previously admitted? Do they have chronic illnesses? Etc.

Lastly we do data splitting! We do this to train and test data. We apply the code attached.

We address over fitting by regularization. This adds a penalty to the given model, this is to avoid having a large coefficient. Regularization is used in logistic regression and Neural Networks. This helps balance the bias.

# Part 4: Reflection & Workflow Diagram

1. What was the most challenging part of the workflow? Why?

The most challenging part of the workflow was bridging the gap between the clean, theoretical model and the messy, real-world deployment. Specifically, this involves two key stages:

- Data Strategy & Preprocessing: In a real-world scenario like the hospital case study, data is never ready to use. It comes from different sources (EHRs, lab systems), in different formats, with countless missing values, and potential biases. Defining a robust preprocessing pipeline that can handle this complexity consistently, while also creating meaningful features (feature engineering), is difficult because any error here directly corrupts the entire model. It's unglamorous but absolutely critical work.

- Deployment & Compliance: Building a accurate model is one thing; integrating it into a complex, secure, and highly regulated environment like a hospital is another. Ensuring compliance with regulations like HIPAA adds a significant layer of complexity, requiring careful planning for data anonymization, secure APIs, and audit trails. This challenge is less about pure data science and more about systems engineering and ethical/legal responsibility.

2. How would you improve your approach with more time/resources?

With more time and resources, I would focus on three key improvements:

· Advanced Feature Engineering & MLOps: I would invest more time in automated feature engineering tools and conduct deeper exploratory data analysis to uncover non-obvious predictive factors. Furthermore, I would implement a full MLOps (Machine Learning Operations) pipeline using tools like MLflow or Kubeflow. This would automate the entire workflow—from data ingestion and retraining to deployment and monitoring—making the system more robust, scalable, and self-healing in production.

· Enhanced Explainability (XAI): I would integrate advanced Explainable AI (XAI) techniques, such as SHAP (SHapley Additive exPlanations), into the deployed model. This would allow doctors to not just see a risk score but also understand why the model assigned that score (e.g., "high risk due to recent medication X and age Y"). This builds trust and facilitates

better clinical decision-making.· Proactive Bias Mitigation and Monitoring: Instead of a one-time bias check, I would
implement continuous bias monitoring as part of the MLOps pipeline. Using tools like IBM AIF360, I would continuously track metrics for fairness across different demographic groups and set up alerts for concept drift and performance degradation, ensuring the model remains fair and effective over time.

Diagram (5 points): AI Development Workflow
The following flowchart provides a clear, high-level overview of the end-to-end AI development lifecycle for a project like the hospital readmission predictor.

```mermaid
flowchart TD
subgraph A [Phase I: Planning & Scoping]
direction LR
A1[Problem Definition] --> A2[Define Objectives & KPIs];
end
subgraph B [Phase II: Data Preparation]
B1[Data Collection] --> B2[Data Preprocessing<br>Cleaning & Transformation];
B2 --> B3[Feature Engineering];
end
subgraph C [Phase III: Modeling & Evaluation]
C1[Model Selection & Training] --> C2[Model Evaluation<br>Metrics & Validation];
C2 --> C3{Performance<br>Acceptable?};
C3 -- No --> C1;
end
subgraph D [Phase IV: Deployment & Monitoring]
D1[Deployment<br>Integration & API] --> D2[Live Monitoring<br>Performance & Drift];
D2 --> D3[Maintenance & Retraining];
end
A --> B;
B --> C;
```

C -- Yes --> D;

linkStyle 3 stroke:green,stroke-width:2px,color:green;

```

Diagram Explanation:

The workflow is a cyclical process, not a linear one, emphasizing continuous improvement.

**Phase I:** Planning & Scoping: This is the foundation. Without a clear problem and success metrics, the project lacks direction.

**Phase II**: Data Preparation: Often the most time-consuming phase. It involves gathering data, cleaning it, and creating informative features for the model to learn from.

**Phase III**: Modeling & Evaluation: This is the iterative core where models are built, tested, and refined until they meet the performance standards defined in Phase I. The loop back from evaluation to training is crucial.· Phase IV: Deployment & Monitoring: The model is integrated into the real-world system.

Continuous monitoring is essential to ensure it performs as expected over time, leading to maintenance and potential retraining, which closes the loop back to the data and modeling phases.
```