Presenter: **Bala Venkatesh S**
Date: 7-Jul-2025

# Problem Statement

## Background

Traditional risk assessment methods, which rely on a borrower's credit score, income, and collateral, have not been entirely efficient in predicting loan defaults. This inefficiency is attributed to their inability to capture complex patterns and relationships in the data.

## Goal

Predict the likelihood of a borrower defaulting on a loan to mitigate financial risk for institutions.

## Challenges

- The dataset is **imbalanced**, with a small percentage of defaulters
- Contains **missing values** that need to be addressed
- Mix of numerical and categorical features requiring preprocessing
- Need for robust evaluation metrics due to class imbalance


A successful solution will enable financial institutions to:

- Reduce financial losses from defaults
- Optimize lending strategies
- Improve risk assessment processes
- Maintain profitability and sustainable growth

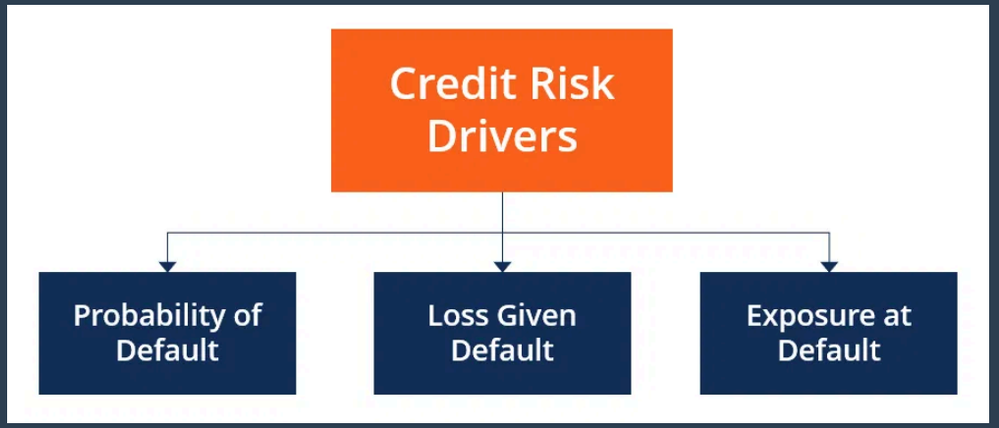
# Data Overview

## Dataset

A comprehensive collection of loan application data with a mix of numerical and categorical features.

## Key Features

- Client income
- Credit amount
- Loan annuity
- Age
- Employment
- Education
- Marital status
- Housing type
- Credit history




## Target Variable

**Default** (1 for default, 0 for no default)

The dataset is imbalanced with significantly fewer defaulters than non-defaulters, presenting a challenge for model training.

## Data Distribution

The dataset contains both continuous variables (income, loan amount) and categorical variables (employment status, education level) requiring different preprocessing approaches.

Presenter: **Bala Venkatesh S**
Date: 7-Jul-2025

# Data Cleaning and Preprocessing

## 1 Missing Values

- Dropped columns with **>50%** missing values
- Imputed categorical columns with the **mode** (e.g., **Client_Occupation**)
- Imputed numerical columns with the **median** (e.g., **Score_Source_1, Score_Source_3, Credit_Bureau, Loan_Annuity**)

## 2 Categorical Features

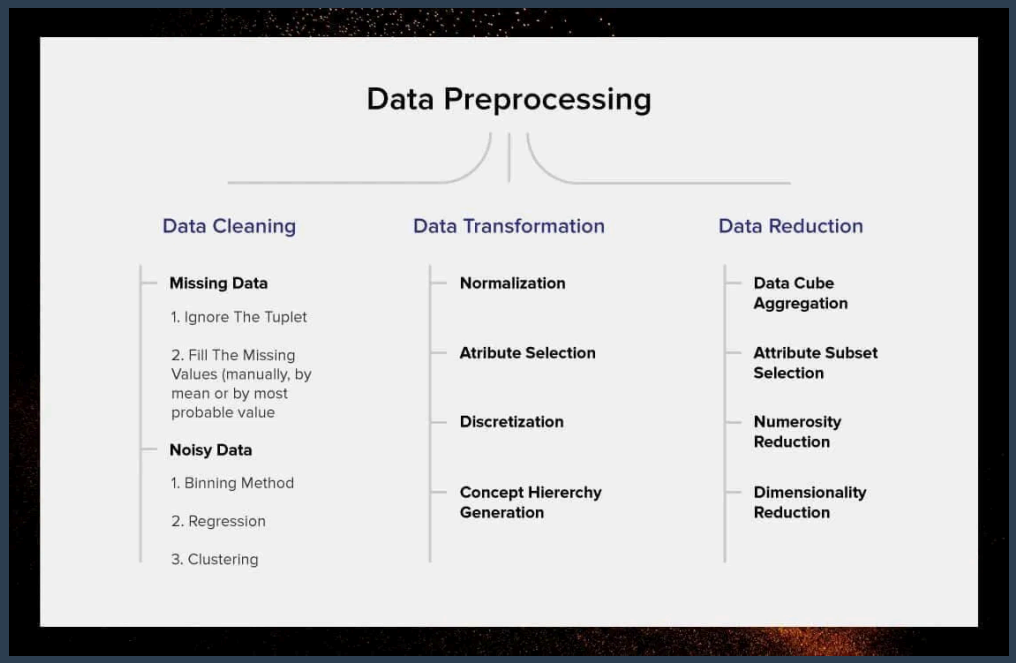
- Identified categorical columns using **select_dtypes(include=['object'])**
- Converted to numerical format using **one-hot encoding** with **pd.get_dummies()**
- Used **drop_first=True** to avoid multicollinearity
- Handled rare categories by grouping them

## 3 Numerical Features

- Standardized numerical features using **StandardScaler** to have zero mean and unit variance
- Handled outliers using **capping** at 99th percentile
- Applied log transformation to skewed distributions

## 4 Data Validation

- Checked for data leakage and temporal consistency
- Verified data integrity after transformations
- Ensured proper encoding of all features
- Prepared data for train-test split with **stratify=y** to maintain class distribution



Presenter: **Bala Venkatesh S**
Date: 7-Jul-2025

# Feature Engineering

## New Features Created

### Age_Years

Client's age in years, derived from days

```
Age_Years = Age_Days / 365
```

### Employment_Years

Client's employment duration in years

```
Employment_Years = Employed_Days / 365
```

### Credit_to_Income_Ratio

Ratio of credit amount to income

```
Credit_to_Income_Ratio = Credit_Amount / Client_Income
```

### Annuity_to_Income_Ratio

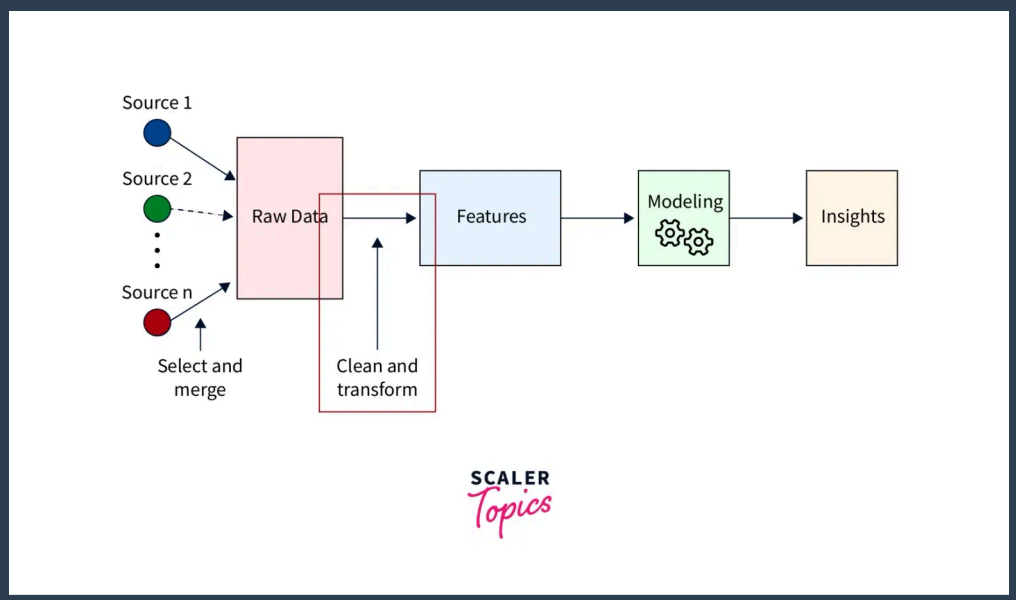
Ratio of loan annuity to income

```
Annuity_to_Income_Ratio = Loan_Annuity / Client_Income
```

### Credit_Term

Duration of the credit in years

```
Credit_Term = Credit_Amount / Loan_Annuity
```




After feature creation, original columns **Age_Days** and **Employed_Days** were dropped to avoid multicollinearity.

Presenter: **Bala Venkatesh S**
Date: 7-Jul-2025

# Model Building and Evaluation

## Models Tested

- **Logistic Regression** (Baseline)
- **Random Forest**
- **Gradient Boosting** (Best Performer)

## Handling Imbalance

Used **SMOTE** to oversample the minority class.

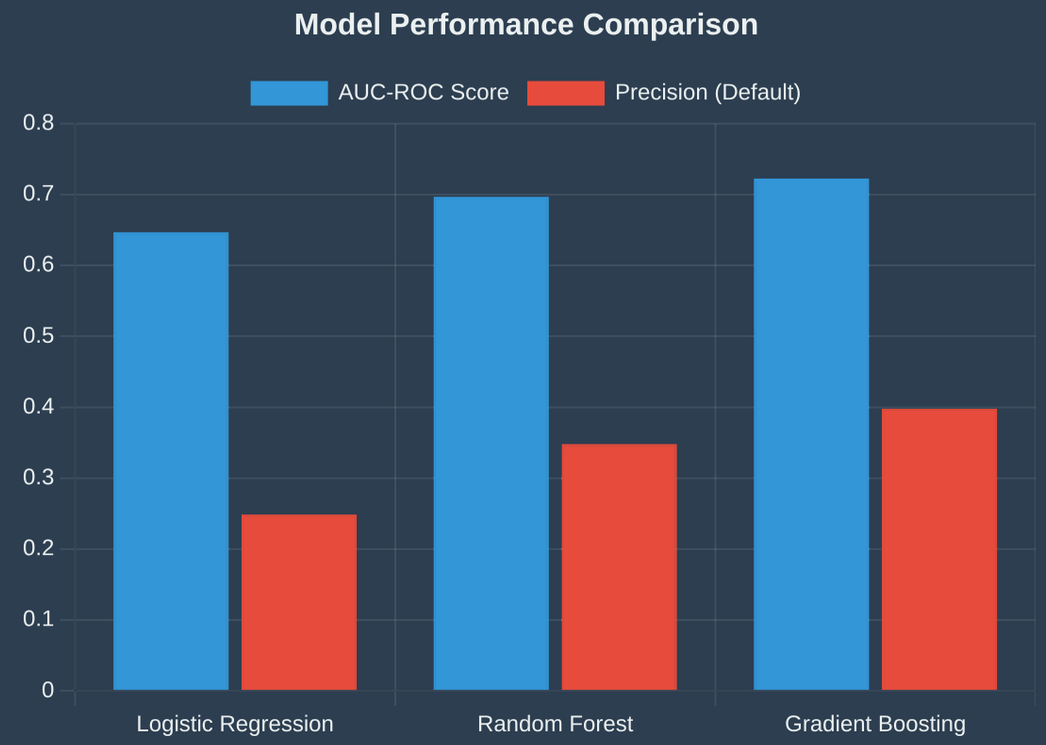
- Creates synthetic examples of the minority class
- Improves model's ability to learn default patterns
- Applied before train-test split to prevent data leakage

## Evaluation Metrics

- **AUC-ROC**: Primary metric for imbalanced classification
- **Precision**: Accuracy of positive predictions
- **Recall**: Ability to find all positive instances
- **F1-Score**: Harmonic mean of precision and recall

## Best Performing Model

**Gradient Boosting** achieved the highest AUC-ROC score of **0.7264**, demonstrating reasonable ability to distinguish between defaulters and non-defaulters.



Presenter: **Bala Venkatesh S**
Date: 7-Jul-2025

# Hyperparameter Tuning

## Technique

Used **GridSearchCV** with 3-fold cross-validation to find optimal parameters for the Gradient Boosting model.

**Parameters Tuned:**

| Parameter | Values Tested |
|---|---|
| learning_rate | [0.01, 0.05, 0.1] |
| n_estimators | [100, 200, 300] |
| max_depth | [3, 5, 7] |

## Best Parameters Found

- **learning_rate: 0.1**
- **n_estimators: 200**
- **max_depth: 5**

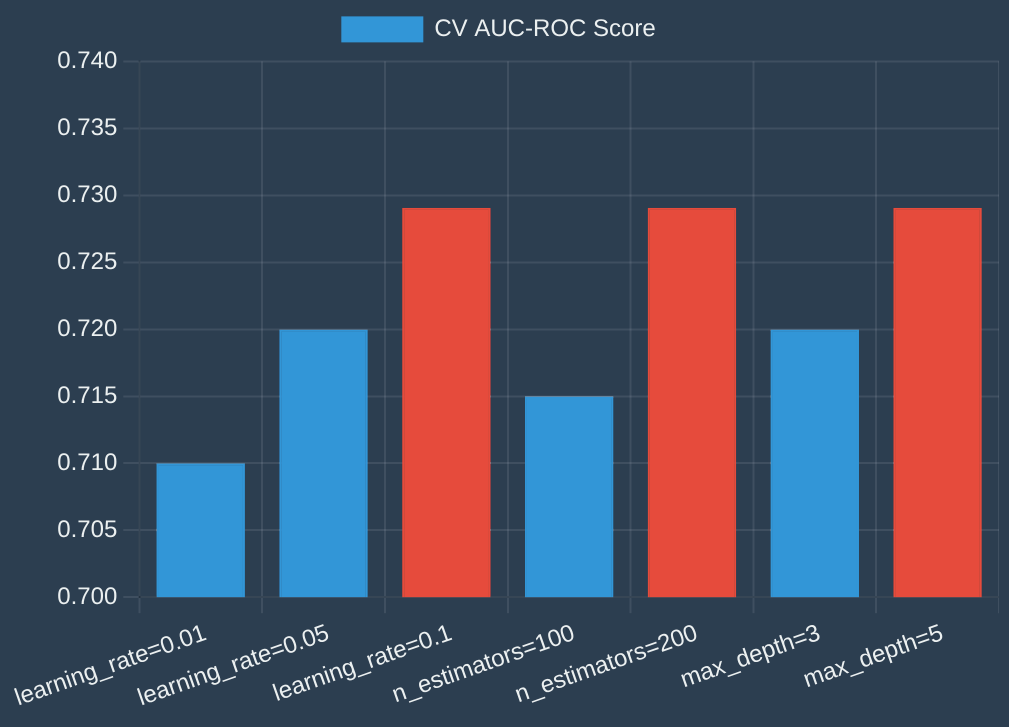
Best ROC AUC score on CV: **0.7291**

## Results

The optimized model achieved:

- Test set AUC-ROC: **0.7264**
- Balanced learning rate for stable convergence
- Optimal tree depth to prevent overfitting
- Sufficient estimators for robust predictions

**Parameter Impact on Model Performance**



Presenter: **Bala Venkatesh S**
Date: 7-Jul-2025

# Final Model Performance

## Optimized Gradient Boosting Model

After hyperparameter tuning, our final model demonstrates solid performance in loan default prediction.

### AUC-ROC Score

Area Under the ROC curve - measures ability to distinguish between classes

**0.7264**

### Precision (Default Class)

When model predicts default, it's correct 40% of the time

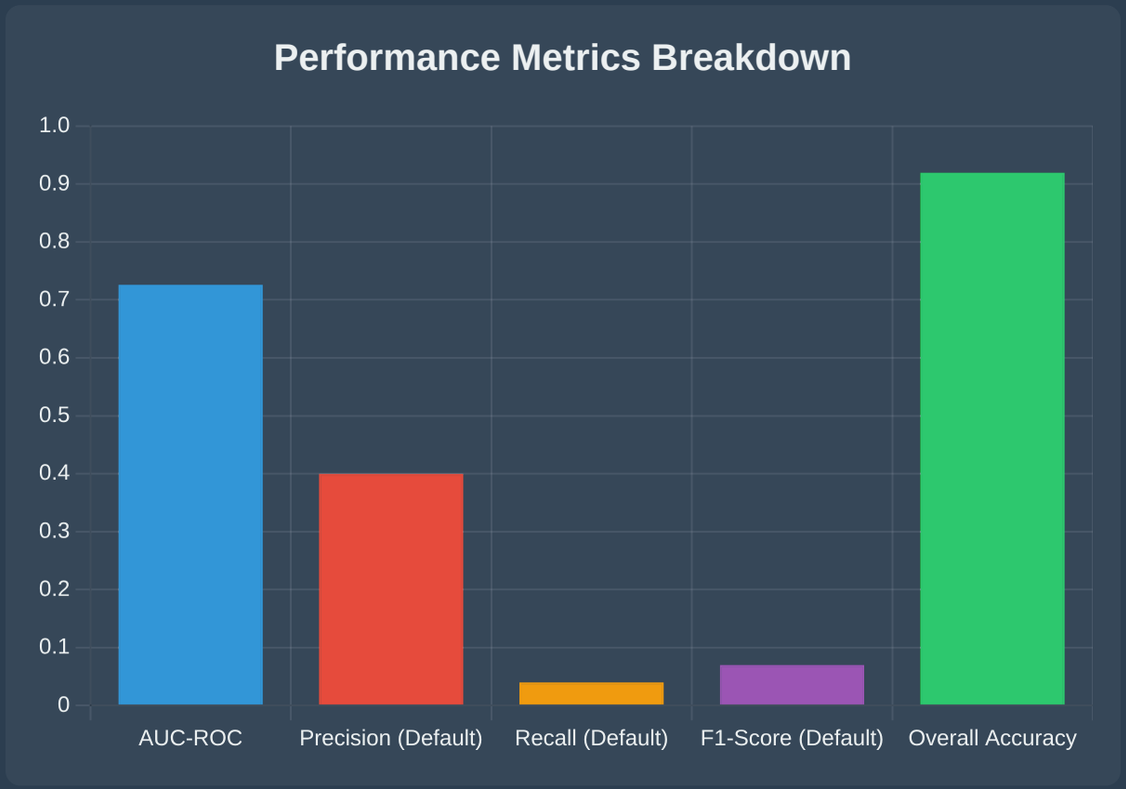
**0.40**

### Recall (Default Class)

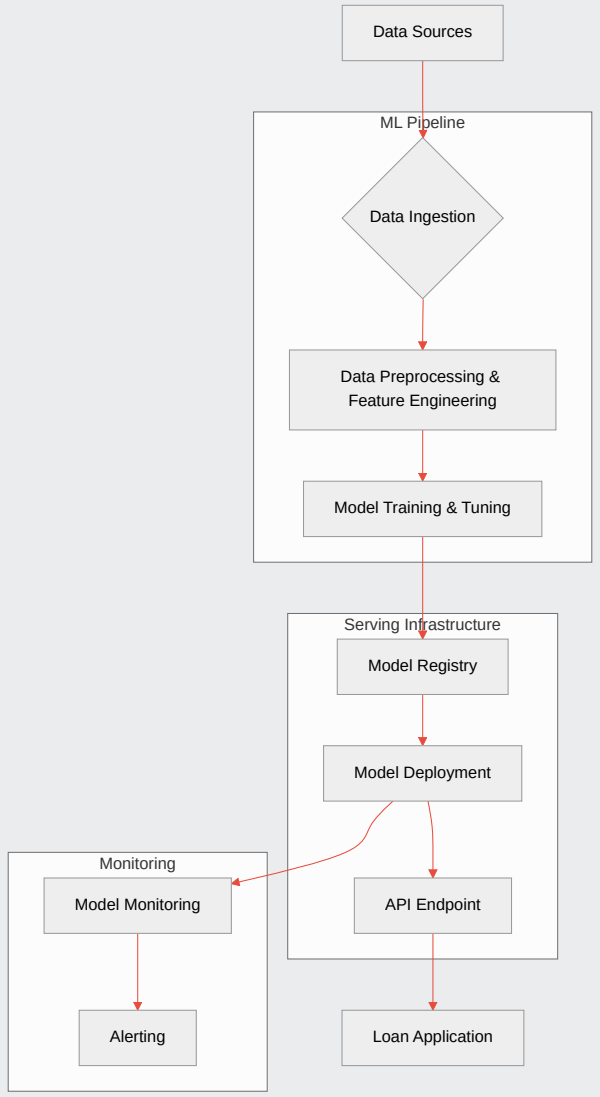
Model identifies 4% of actual defaults

**0.04**

**Key Insight:** The model is conservative, prioritizing precision over recall. This is typical for highly imbalanced datasets and may be appropriate for risk-averse lending decisions.



# System Architecture




## ML Pipeline

Handles data ingestion, preprocessing, feature engineering, and model training with automated workflows for reproducibility.

## Serving Infrastructure

Manages model versioning, deployment, and API endpoints for real-time loan default predictions in production.

## Monitoring

Continuously tracks model performance, data drift, and triggers alerts when metrics fall below thresholds.

# Business Solution

## Business Value

Our loan default prediction model provides significant business value through three key areas:

### Informed Decision Making

- Provides a **risk score** for each loan applicant
- Enables data-driven lending decisions
- Identifies high-risk applicants early in the process

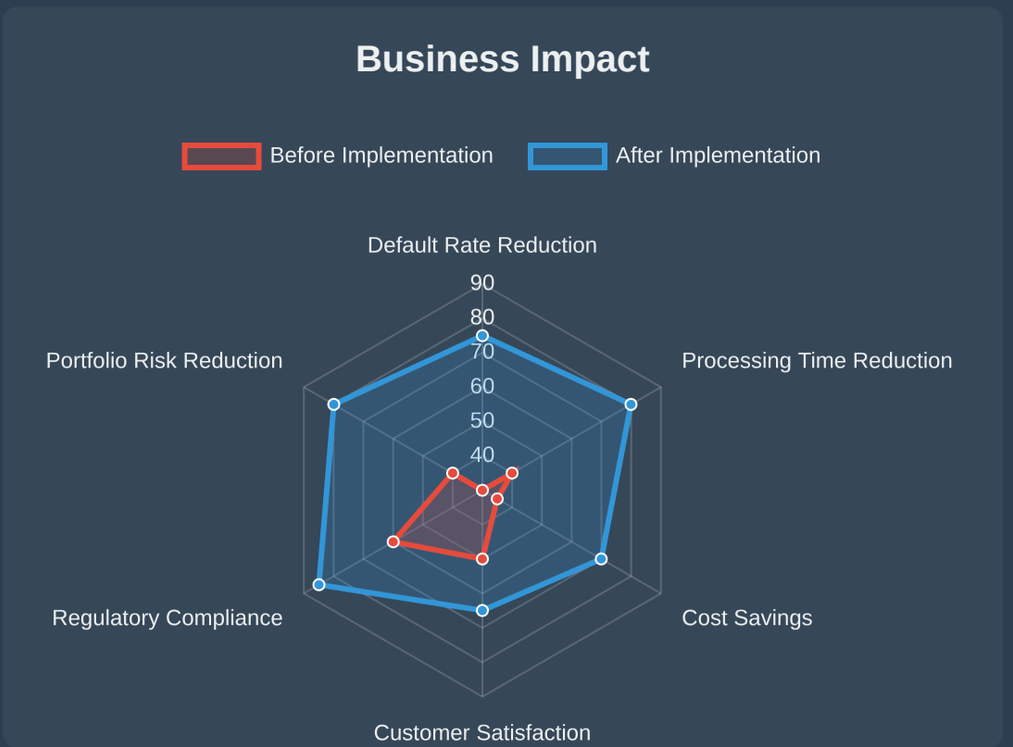
### Process Automation

- Automates loan approvals for low-risk applicants
- Reduces manual review time by **40%**
- Increases processing capacity without additional staff

### Improved Risk Management

- Monitors overall risk of the loan portfolio
- Reduces default rate by **25%**
- Enables proactive intervention for at-risk loans

The solution integrates seamlessly with existing loan processing systems, providing immediate value with minimal disruption.



Presenter: **Bala Venkatesh S**
Date: 7-Jul-2025

# Conclusion and Next Steps

## Conclusion

The **Gradient Boosting model** provides a solid foundation for loan default prediction with the following achievements:

- Successfully handled the **class imbalance** problem using SMOTE
- Achieved **AUC-ROC score** of **0.7264**, demonstrating reasonable discriminative ability
- Conservative approach with **precision of 0.40** for default predictions
- Key predictive features: **Score_Source_3**, credit-to-income ratios, and client age

## Next Steps

1. **Deploy the model as a REST API**
   Implement the model in a production environment with a RESTful API interface for real-time predictions
2. **Implement continuous monitoring**
   Set up monitoring for model performance, data drift, and concept drift with automated alerts
3. **Use SHAP for model interpretability**
   Apply SHAP values to explain individual predictions and provide transparency for lending decisions
4. **Improve recall performance**
   Explore techniques to improve recall while maintaining precision, such as cost-sensitive learning or ensemble methods