FEE PAYMENT DEFAULT PREDICTION USING MACHINE LEARNING

ABSTRACT

Educational institutes face challenges in identifying students who are likely to delay or default on fee payments. Manual tracking is time-consuming, error-prone, and often detects defaulters late in the process.

This project builds an automated Fee Payment Default Prediction System using machine learning that forecasts:

- Whether a student will delay fee payment
- Whether the student needs a fee reminder

Using historical ERP data, the model learns patterns in fee structure, payment behaviour, course type, and admission status. After preprocessing and model training, the solution was deployed into a Streamlit web application integrated with a rule-based engine to ensure higher practical accuracy.

This system helps educational institutions automate decision-making, improve fee collection, and identify at-risk students early.

Introduction

1. Problem Statement:

Institutes need an automated mechanism to:

- Identify students who are at risk of delayed fee payment
- Detect students who require reminders based on payment patterns
- Predict fee behaviour using financial and academic indicators
- Reduce manual follow-ups and improve fee recovery workflow

Traditional systems rely on staff monitoring or end-term verification, which delays interventions.

A predictive system can provide early warnings and automate follow-up actions.

2. Objective:

- Build ML models to predict Delayed Payment and Reminder Need
- Perform detailed Exploratory Data Analysis (EDA)
- Clean and preprocess ERP dataset
- Train and evaluate multiple machine learning models
- Integrate model + rule engine inside a Streamlit web app
- Provide institutes with real-time fee-risk predictions

Dataset Description

Dataset Name: Educatinal_ERP_Dataset
Records: ~**5000**
Data Sheets Used: **Student_Admission & Fee_Collection**

Features:

Feature	Description
Admission_status	Whether the student is admitted
Total_fee	Total course fees
Paid_fee	Fee amount paid
Balance	Auto-calculated: *total_fee – paid_fee*
Course	Course enrolled
Needs_reminder	Target variable (Yes/No)
Delayed_payment	Target variable (Yes/No)

Target Variables:

ML Tasks	Target Variables
Delayed Payment Prediction	**YES/NO**
Needs Reminder Prediction	**YES/NO**

Exploratory Data Analysis (EDA)

EDA was carried out using Pandas, Matplotlib, and Seaborn.

Key Insights

- Students who pay less than 70% of the total fee have a high chance of needing reminders.
- Admission confirmation strongly influences fee payment behaviour.
- Course type shows varying fee patterns depending on stream.
- Balance amount is the strongest indicator of delayed payment.
- Outliers were present in fee values and were handled appropriately**.**

Visualizations Included

- Correlation Heatmap
- Fee Distribution Plot
- Box Plot
- Admission vs Fee Payment Analysis
- Balance vs Delayed Payment Plot

Data Preprocessing

Steps Applied

1. **Handling Missing Values**
   - Numerical columns → Mean/Median
   - Categorical columns → Most Frequent
2. **Encoding**
   - Label Encoding for categorical features (course, admission status, etc.)
   - Stored encoders in label_encoders.pkl
3. **Feature Scaling**
   - StandardScaler used for numerical variables

4. **Feature Engineering**
   - Created balance = total_fee - paid_fee
   - Derived payment ratio for logic layer
5. **Train-Test Split**
   - 80% training
   - 20% testing

Output Files Generated:

- scaler.pkl
- feature_columns.pkl
- label_encoders.pkl
- best_model_delayed_payment.pkl
- best_model_needs_reiminder.pkl

Model Training

Two ML models were built:

1. Delayed Payment Prediction

- **Algorithm:** RandomForestClassifier
- Evaluated against Decision Tree & Logistic Regression
- Achieved high accuracy on predicting late payers

2. Reminder Prediction

- **Algorithm:** RandomForestClassifier
- Predicts if student needs a fee reminder
- Also combined with **70% rule engine** for accuracy:

Business Rule Applied

If:
paid_fee / total_fee <= 0.70 → Needs Reminder = Yes

This improved real-world reliability.

Model Evaluation

Metrics Used

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix

Results

Model	Accuracy
Delayed Payment Model	~99%
Reminder Prediction Model	~99%
Rule Engine + ML Output	100% consistency in fee reminder decisions

Feature Importance

Important predictors include:

- Paid fee amount
- Total Fee Amount
- Balance remaining
- Admission status
- Course type

Fee behaviour trends proved the most influential in predicting both outputs.

Deployment

The final model was integrated into a **Streamlit App (app.py)**.

App Features

- Enter student details
- Auto-calculates balance

- Predicts:
  - **Delayed Payment**
  - **Needs Reminder**
- Built-in **rule engine** for 70% fee check
- Attractive UI + balloons on success
- Models loaded from .pkl files

Conclusion

This Machine Learning solution effectively predicts student fee payment behaviour and enables institutions to:

- Identify fee defaulters early
- Automate reminder notifications
- Reduce manual administrative work
- Improve total fee recovery
- Integrate data-driven decision-making

The combined model + rule engine system makes predictions practical, accurate, and deployment-ready.

Future Scope

- Add automatic SMS/WhatsApp reminders
- Integrate with payment gateway
- Add dropout prediction model
- Build mobile-app version
- Include fee forecasting for full academic year
- Add dashboards for administrators