Project Proposal: Deployment of a Student Dropout Prediction System

Name: *PIYUSH UKEY*
Registration Number: *2024SEPVUGP0044*
Course: Hackathon 3 – Development of Pipelines and Maintenance of Models

1. Abstract

This project proposes the development and deployment of a classification-based system to predict the likelihood of student dropout in a university setting. Synthetic but realistic student academic and socio-economic data is generated using Python scripts and stored in a SQL database to ensure scalability, reproducibility, and maintainability. Traditional machine learning models, specifically Random Forest and Logistic Regression, are trained and evaluated using this data, with primary emphasis on model usability, deployment, and lifecycle management rather than solely on performance optimization.

A dashboard-based user interface is developed to visualize dataset insights, model performance metrics, and enable real-time dropout risk predictions based on user-provided student information. The system is designed with a well-defined maintenance strategy that supports periodic data updates, model retraining, versioning, and seamless integration with the prediction dashboard.

2. Problem Statement

Student dropout is a significant challenge faced by universities, leading to academic, financial, and administrative consequences. Dropout rates are influenced by multiple factors such as attendance, family income, financial support (scholarships), residential status (hostel/day scholar), and previous academic performance. Identifying students at risk of dropping out at an early stage can help universities implement timely interventions, provide counseling, financial aid, or academic support.

The objective of this project is to predict whether a student is **likely to drop out or continue** using historical student data and traditional machine learning classification models. The focus of the project is on building a deployable, maintainable, and reproducible machine learning pipeline rather than achieving the highest possible model accuracy.

3. Data Description

The dataset used in this project is synthetically generated using a Python-based data synthesis script to simulate realistic student records within a university environment. The synthetic data allows reproducibility, controlled experimentation, and avoids dependence on static or sensitive real-world student data.

All generated data is stored in a structured SQL database (SQLite), which serves as the central and authoritative data source for the project. New student data can be appended periodically to simulate real-world data growth and support continuous model retraining.

Input Features

- student_id
- attendance (%)
- family_income (INR)
- hostel (binary: 1 = Hostel resident, 0 = Day scholar)
- scholarship (binary: 1 = Yes, 0 = No)
- previous_gpa

Target Variable

- dropout (binary: 1 = Dropped out, 0 = Continued)

4. Model Implementation and Evaluation

In alignment with Hackathon 3 guidelines, traditional machine learning models are used instead of deep learning approaches. Two classification models are implemented and compared:

- **Logistic Regression**
- **Random Forest Classifier**

Model training is performed using data directly retrieved from the SQL database to ensure integration between data storage and model development. The dataset is split into training and testing sets to evaluate generalization performance.

Model performance is assessed using standard classification metrics including:

- Accuracy
- Precision
- Recall
- Confusion Matrix

The comparison of these metrics is used to select the most suitable model for deployment, prioritizing stability, interpretability, and usability over marginal performance gains.

5. Prediction Readiness and Dashboard

The trained model is made prediction-ready by serializing it using Python's pickle module and integrating it into a Streamlit-based dashboard. The dashboard serves as a user-friendly interface that allows university administrators, faculty, or counselors to input student details and receive real-time dropout risk predictions.

The dashboard provides:

- A preview of the dataset stored in SQL
- Summary statistics and basic visual analysis
- Model performance results
- An interactive input form for prediction
- Clear output: **"High Dropout Risk"** or **"Low Dropout Risk"**

The dashboard is designed to always load the **latest trained model version**, ensuring that predictions reflect the most recent training cycle.

6. Model Update and Maintenance Timeline

The project follows a structured model lifecycle management approach:

- New student data will be periodically added to the SQL database to simulate real-world data accumulation.
- Model retraining will be performed after a predefined interval (e.g., every semester or after significant data updates).
- Each retraining cycle will generate a new version of the model (e.g., dropout_model_v1.pkl, dropout_model_v2.pkl).
- Older model versions will be preserved for comparison, auditing, and reproducibility.
- The dashboard will automatically load and use the most recent model version for predictions.

7. Version Control and Reproducibility

A public GitHub repository is maintained for this project to ensure transparency, collaboration, and reproducibility. The repository follows a clean and modular folder structure separating:

- Data generation scripts
- Database handling scripts
- Model training and retraining scripts
- Prediction logic

- Dashboard/UI code

Commit history clearly reflects incremental project development, including:

- Initial project setup
- Data pipeline creation
- Model training implementation
- Dashboard development
- Model retraining updates

Large datasets and trained model binaries are excluded from version control to maintain repository efficiency. Instead, data generation scripts are included to allow complete reproducibility of the project.

8. Expected Output

By the end of the project, the following deliverables will be achieved:

A fully functional SQL-backed data storage pipeline

Trained and evaluated machine learning models for dropout prediction

A prediction-ready system capable of handling new student inputs

An interactive Streamlit dashboard for analysis and real-time prediction

A maintainable and scalable machine learning pipeline with a clear retraining strategy

A publicly accessible GitHub repository with well-documented commit history