Machine Learning
Individual Assignment

DEBRE BIRHAN UNIVERSITY
COLLEGE OF COMPUTING
DEPARTMENT OF SOFTWARE ENGINEERING

Set by - Elsa Abera……..1402648

Sebmited to -Derbew Felasman

Student Dropout Prediction - Personalized Machine Learning Project

Problem Definition

Student dropout is a significant issue in the education sector, affecting institutions, students, and society as a whole. The reasons for student dropout can vary widely, including academic struggles, financial difficulties, lack of engagement, and personal challenges. Educational institutions face the challenge of identifying students at risk early enough to provide necessary interventions.

The consequences of student dropout are far-reaching, leading to financial loss for institutions, reduced workforce potential, and adverse psychological effects on students. By leveraging machine learning techniques, we aim to identify patterns in student data that correlate with dropout tendencies. A well-trained predictive model can assist educators and policymakers in making informed decisions to support at-risk students and improve retention rates.

Objectives

- **Develop a predictive model** to identify students at risk of dropping out based on various academic, socio-economic, and behavioral features.
- **Perform Exploratory Data Analysis (EDA)** to uncover patterns, trends, and insights from the dataset, helping in feature selection and understanding underlying dropout causes.
- **Preprocess the dataset** by handling missing values, standardizing variables, encoding categorical features, and addressing data imbalances.
- **Select and compare multiple machine learning algorithms** such as Logistic Regression, Decision Trees, Random Forest, Support Vector Machines, and Neural Networks to determine the best-performing model.
- **Optimize model performance** by using techniques like hyperparameter tuning, cross-validation, and feature engineering to improve prediction accuracy.
- **Evaluate the model using robust metrics**, including accuracy, precision, recall, F1-score, and AUC, to assess its effectiveness in predicting student dropout.
- **Visualize key findings** through graphs and charts to enhance interpretability and provide a clear understanding of influential features.
- **Compare the model's performance against a baseline classifier**, such as a simple rule-based system, to establish its practical utility.
- **Deploy the trained model as an API** using FastAPI, allowing real-time predictions for new student data and enabling integration with existing education management systems.
- **Provide thorough documentation**, including a structured report, well-commented code, and a clear README file, to ensure reproducibility and ease of understanding for future users and researchers.

Data Source and Description

The dataset used for this project contains information on students, including academic performance, attendance records, socio-economic background, and personal attributes. The dataset includes:

- **Demographic Information**: Age, gender, nationality, parental education level, etc.
- **Academic Performance**: Grades, GPA, course completion rate, exam scores.
- **Behavioral Data**: Attendance, engagement levels, disciplinary records.
- **Financial and Socio-Economic Factors**: Scholarship status, financial aid, employment status.
- **Dropout Label**: Whether the student eventually dropped out or continued their studies.

Exploratory Data Analysis (EDA)

EDA is performed to gain insights into the dataset before training the model. This includes:

- Checking for missing values and handling them appropriately.
- Analyzing the distribution of key features using histograms and box plots.
- Identifying correlations between features and dropout rates using correlation matrices.
- Visualizing class distributions to detect any imbalances in the dataset.

Data Preprocessing

- **Handling Missing Values**: Imputation techniques applied where necessary.
- **Standardizing and Normalizing Features**: Ensuring consistency across numerical attributes.
- **Encoding Categorical Variables**: Using one-hot encoding or label encoding.
- **Feature Selection**: Removing irrelevant or redundant features.
- **Splitting the Dataset**: Dividing data into training (80%) and testing (20%) sets.

Model Implementation and Training

1. **Algorithm Selection**: Based on performance metrics, we select models such as Logistic Regression, Decision Trees, Random Forest, or Neural Networks.
2. **Training the Model**: Using the preprocessed training data.
3. **Hyperparameter Tuning**: Applying cross-validation techniques to optimize model performance.
4. **Handling Class Imbalance**: Using techniques like SMOTE or class weighting if necessary.

Model Evaluation and Analysis

- **Predictions on Test Data**: Using the trained model to classify students.
- **Performance Metrics**:
    - Accuracy, Precision, Recall, F1-score, and AUC for classification models.
    - Comparison against a baseline classifier.

- **Visualizing Results**:
    - Confusion matrix, ROC curves, and feature importance graphs.
- **Interpretation of Metrics**: Understanding model strengths and limitations.

Model Deployment

- **Deploying as an API using FastAPI**: Making the model accessible for real-time predictions.
- **API Functionality**:
    - Accepting student data as input.
    - Returning dropout probability as output.
- **Deployment Instructions**:
    - Setting up a FastAPI server.
    - Running the API locally or deploying it to cloud platforms.
    - Testing API endpoints with sample data.

Potential Limitations and Future Improvements

- **Data Quality Issues**: Limited or biased datasets can impact accuracy.
- **Generalizability**: Model may not perform well on unseen institutions.
- **Feature Expansion**: Additional student behavior data could improve predictions.
- **Real-time Monitoring**: Implementing continuous learning for adapting to new patterns.

Documentation and Reproducibility

- **Well-commented code** for clarity and maintainability.
- **Detailed report** covering:
    - Problem statement, data analysis, preprocessing, model selection, evaluation, deployment.
- **README file** providing clear instructions on:
    - Setting up the environment.
    - Running the model and API.
    - Understanding the results and outputs.