**Student Dropout Prediction Web Application Documentation**

**Table of Contents**

1. [Introduction](#introduction)
2. [Features](#features)
3. [Installation](#installation)
4. [Usage](#usage)
   * [User Guide](#user-guide)
   * [Examples](#examples)
5. [Technical Details](#technical-details)
6. [Model Limitations](#model-limitations)
7. [Use Cases](#use-cases)
8. [Code Structure](#code-structure)
9. [Contributing](#contributing)
10. [License](#license)

**Introduction**

The **Student Dropout Prediction Web Application** is a machine learning-based tool designed to predict the likelihood of student dropout. By analyzing various factors such as academic performance and demographic details, the application provides insights into students' dropout risks. It utilizes advanced machine learning techniques, including SHAP for feature importance analysis, to help educators and administrators make informed decisions.

**Features**

* **User-Friendly Interface**: Simple forms for entering student information.
* **Dropout Prediction**: Provides predictions based on user inputs and the underlying machine learning model.
* **Feature Importance Analysis**: Displays the significance of different features in the prediction using SHAP values.
* **Interactive Visualizations**: Features plots that help users understand the model's decisions.

**Installation**

To set up the Student Dropout Prediction Web Application, follow these steps:

**Prerequisites**

* Python 3.x
* pip (Python package installer)

**Step-by-Step Installation**

1. **Clone the Repository**:

bash

Copy code

git clone <repository-url>

cd dropout\_prediction\_app

1. **Install Required Packages**: Create a virtual environment (optional but recommended):

bash

Copy code

python -m venv venv

source venv/bin/activate # On Windows use `venv\Scripts\activate`

Install the necessary packages:

bash

Copy code

pip install -r requirements.txt

1. **Run the Application**:

bash

Copy code

streamlit run app.py

1. **Access the Application**: Open your web browser and go to http://localhost:8501.

**Usage**

**User Guide**

1. **Entering Data**:
   * **Categorical Inputs**: Use the dropdown menus to select values for categorical features like Marital Status and Previous Qualification.
   * **Numerical Inputs**: Input values for numerical features using the number input fields. For example, for Admission Grade, enter the numeric grade based on previous performance.
2. **Making Predictions**:
   * After filling in the details, click the **"Predict"** button to receive the prediction results.
   * The application will display the dropout risk percentage along with a message indicating whether the risk is high or low.
3. **Viewing Feature Importance**:
   * Navigate to the **"Feature Importance"** section to view:
     + **Global Feature Importance**: Summary plots showing how each feature impacts predictions.
     + **Local Explanation**: SHAP force plots that detail the contributions of each feature for the specific prediction.

**Examples**

* **Example Input**:
  + Marital Status: **Single**
  + Previous Qualification: **High School**
  + Age at Enrollment: **20**
  + Admission Grade: **85**
* **Example Output**:
  + **Dropout Risk**: **65.5%**
  + Message: **High Dropout Risk**

This example illustrates a scenario where a student with a high risk of dropout has been identified, prompting the need for intervention.

**Technical Details**

**Model and Libraries**

* The application leverages a machine learning model (e.g., Logistic Regression, Random Forest) trained on historical student data.
* Key libraries used include:
  + **Streamlit**: For building the web interface.
  + **Pandas**: For data manipulation and handling.
  + **Scikit-learn**: For implementing machine learning algorithms.
  + **SHAP**: For explaining model predictions and understanding feature importance.

**Model Limitations**

While the Student Dropout Prediction model provides valuable insights, it has several limitations:

* **Data Dependence**: The model's accuracy is highly dependent on the quality and representativeness of the training data. The predictions may be inaccurate if the training data does not capture all relevant factors.
* **Generalization**: The model may not generalize well to students outside the demographic or educational context represented in the training data.
* **Dynamic Factors**: Factors affecting dropout rates can change over time (e.g., economic conditions, educational policies). The model may not adapt to these changes unless retrained with updated data.
* **Interpretability**: Although SHAP values provide insight into feature importance, interpreting complex interactions between features can be challenging.

**Use Cases**

The Student Dropout Prediction application is suitable for various use cases, including:

* **Educational Institutions**: Help colleges and universities identify at-risk students and implement support programs.
* **Policy Makers**: Assist in formulating policies aimed at improving student retention rates based on predictive insights.
* **Research**: Serve as a tool for researchers studying factors influencing student dropout rates.

**Code Structure**

* **app.py**: Main application file that contains Streamlit logic and UI components.
* **model.py**: Contains functions for model training and prediction.
* **utils.py**: Helper functions for data preprocessing and feature engineering.
* **requirements.txt**: Lists all necessary Python packages for the application.

**Contributing**

Contributions to enhance the functionality or features of the Student Dropout Prediction Web Application are welcome. Please follow these steps:

1. Fork the repository.
2. Create a new branch for your feature or fix.
3. Commit your changes and push to your branch.
4. Submit a pull request.

**License**

This project is licensed under the MIT License. See the LICENSE file for more details.