Dropout Insight: Educational Risk Dashboard with Counterfactual Explanations

User Manual

Index

Index of images

1. Introduction

This user manual aims to guide users in using the tool, offering a clear and simple description of its main features.

This document provides the necessary instructions for interacting with the tool, from initial access to executing the various options available.

The overall purpose is to help users install, run, and use the application effectively, as well as to provide solutions to any questions or problems that may arise during use.

This manual has been prepared voluntarily, considering that the tool requires a guide to help people learn how to use it properly.

The document is designed to support both primary users and those with more experience, facilitating understanding of the system through practical explanations and illustrative examples. In addition, recommendations for use and best practice guidelines are included to optimize the experience and make the most of the tool's capabilities.

In this way, users will have a comprehensive and accessible reference to resolve frequently asked questions, understand the operating logic of each module, and use the tool effectively and correctly.

With a clear and easy-to-understand approach, this manual is intended to be an additional, useful, and accessible resource for users, offering them confidence and guidance that will enable them to use the tool successfully.

2. Installation

In order to use the application correctly, it is necessary to follow a series of installation steps that ensure its proper functioning.
Since the entire project is coded in the Python programming language along with various associated libraries, following these specific considerations is important

Recommendations and best practices are included, such as the use of virtual environments, which allow the tool to be kept organized and minimize conflicts with other programs installed on the computer. The tool can only be run in a **local environment**

The purpose of this section is, therefore, to provide the user with a comprehensive and orderly guide so that the installation can be carried out without complications.
The following sections describe step by step the actions necessary to carry out the installation, from preparing the environment to running the application for the first time

The application can be executed in two different ways either using a local python virtual environment or using Docker.

2.1. Prerequesites

Before beginning the installation process, you must check that your system meets the following requirements:

- **Operating system:** Windows 10/11 or recent Linux distributions (e.g., Ubuntu 20.04 or higher).

- **Python**: For the system to work properly, you must install **version 3.9.0 exactly**, due to the compatibility of the different libraries used.

- **Pip:** Updated Python package manager.

- **Web browser**: Google Chrome, Mozilla Firefox, Microsoft Edge, or equivalent updated browser.

- **Internet connection:** To download necessary dependencies and libraries.

- **Docker Desktop or Docker engine**:(optional, only if you prefer running the application in a container)

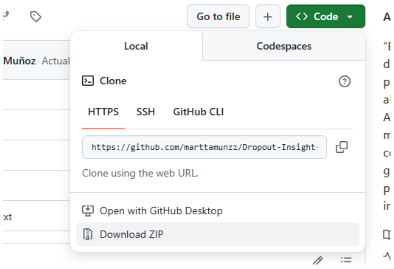
2.2. Code download

Once the prerequisites have been met, the code can be downloaded. To do so, access the GitHub link via the following link: GitHub

For non-technical users, the simplest method is:

- Access the GitHub page
- Press the green "code" button
- Download the project as a .zip file (image 1)
- Unzip the file to your desired directory (this will be the project root folder)

(Note: If you have Git installed, you can use git clone to download the tool to the directory where you are located).



```
https://github.com/marttamunzz/Dropout-Insight-
```

Clone using the web URL.

Image 1: zip download option

2.3. Virtual enviroment creation

It is **strongly recommended to create a virtual environment** to install the dependencies in isolation and avoid conflicts with other programs.

It is important to note that the virtual environment only needs to be created the first time. Once created, each time you want to work with the tool, you only need to activate and deactivate the corresponding environment.

The steps required to create and activate a virtual environment are described below:

- **Open the terminal** (or command line)
- Go to the project's root folder
- Create a virtual environment using the following commands: **python -m venv venv**
- Activate the new created virtual environment:
    - Windows: **venv\Scripts\activate**
    - Linux/MacOS:**source venv/bin/activate**

Once activated, the name of the environment will appear in parentheses at the beginning of the command line, indicating that the virtual environment is in use: **(venv) C:\User\Project>**

When you wish to end the session, you must exit by typing deactivate.

2.4. Installing the required Python libraries

Once the virtual environment has been created and activated, the next step is to install the libraries needed, which are all listed in the **"requirements.txt" file.**

To complete the installation, follow these steps:

- Check that the virtual environment is activated: The name of the environment should appear at the beginning of the command line.
- Run the following command in the project's root folder: **pip install -r requirements.txt**

This command will automatically download and install all the necessary libraries from the official Python repository (PyPI)

- Verify the installation of the libraries by running **pip list**

Important libraries include Dash, Flask, pandas, numpy, scikit-learn, mljar-supervised, lightgbm, xgboost, catboost, explainerdashboard, shap and dice-ml

2.5. Initialize and run the Application

The application is deployed locally using Flask

- Navigate to the source code folder: cd src (or run from the root folder).

- Run the app using the command: **python index.py**

(Note: Running python index.py from the src folder is also possible).

- The application will open automatically in your browser at **http://localhost:8050**

2.6. Running the application with Docker

As an alternative to the local Python installation, the application can be executed using Docker, this option is recommended for users who prefer not to install Python and the required libraries directly on their system

The project includes a Dockerfile located inside the src/ folder.

- To build the image, open a terminal in the project root folder and run: **docker build -t dropout-insight -f src/Dockerfile.**

This command creates a Docker image named dropout-insight that contains the application and all its dependencies.

- Once the image has been built, the application can be started with: **docker run -p 8050:8050 dropout-insight**

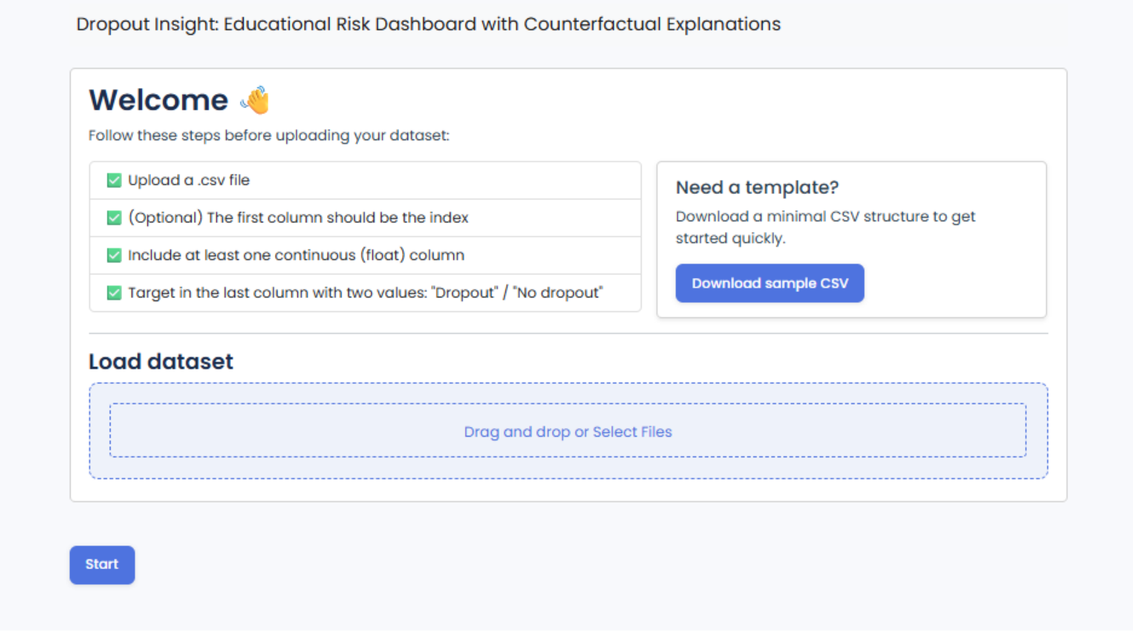
After running the command, the application will be available in the browser at: **http://localhost:8050**

3. Interface

The application interface has been designed to be clear, intuitive, and easy to use, even for users without advanced technical knowledge.

3.1. Main screen and data loading (clear data formatting guidelines)

The home screen displays initial instructions and the option to load the dataset, along with the possibility of **downloading a CSV file template**. This template is highly recommended for ensuring correct data structure.



Dropout Insight: Educational Risk Dashboard with Counterfactual Explanations

Welcome 👋

Follow these steps before uploading your dataset:

- ☑ Upload a .csv file
- ☑ (Optional) The first column should be the index
- ☑ Include at least one continuous (float) column
- ☑ Target in the last column with two values: "Dropout" / "No dropout"

Need a template?

Download a minimal CSV structure to get started quickly.

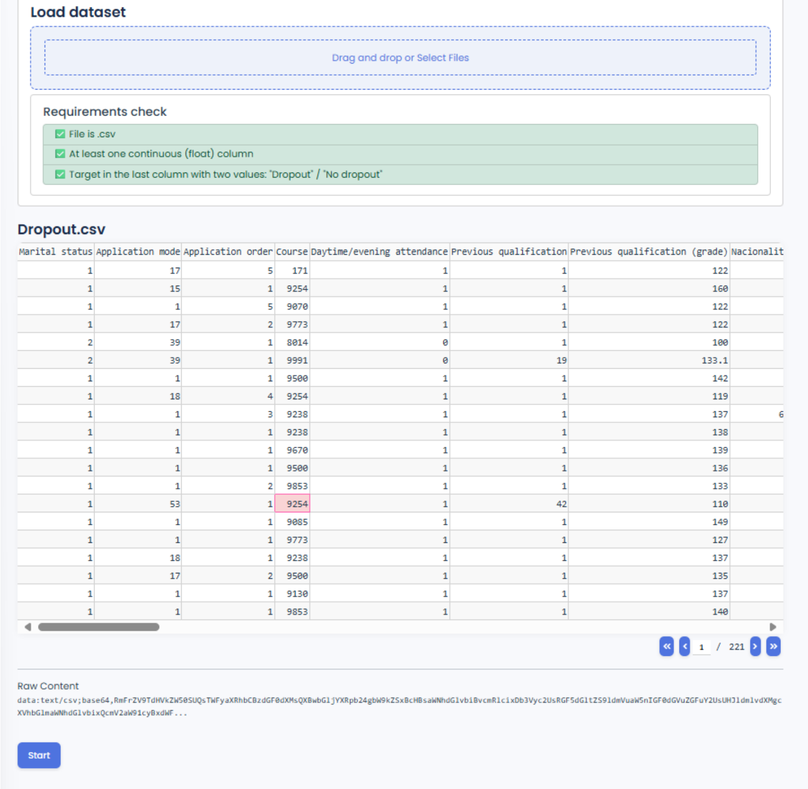
Load dataset

Image 2: Index screen

The tool requires the following conditions to be met for the data analysis to function logically:

- **File Format:** The data file must be in .csv format.
- **Data Upload:** Users can drag and drop the file directly onto the interactive box or manually select it from the file explorer.
- **Verification:** Once the CSV file is uploaded, a table displaying the instances is shown, along with a **checklist** to verify that all specified requirements are met.

The system checks if an already saved model exists. If it doesn't exist, **it is automatically trained from scratch** with the uploaded data and saved. If it exists, the saved model is reused



Load dataset

Requirements check

- ☑ File is .csv
- ☑ At least one continuous (float) column
- ☑ Target in the last column with two values: 'Dropout' / 'No dropout'

Dropout.csv

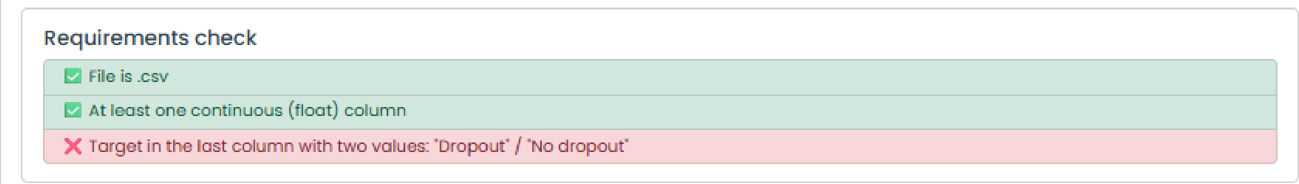
Marital status	Application mode	Application order	Course	Daytime/evening attendance	Previous qualification	Previous qualification (grade)	Nacionalit
1	17	5	171	1	1	122	
1	15	1	9254	1	1	160	
1	1	5	9070	1	1	122	
1	17	2	9773	1	1	122	
2	39	1	8014	0	1	100	
2	39	1	9991	0	19	133.1	
1	1	1	9500	1	1	142	
1	18	4	9254	1	1	119	
1	1	3	9238	1	1	137	6
1	1	1	9238	1	1	138	
1	1	1	9670	1	1	139	
1	1	1	9500	1	1	136	
1	1	2	9853	1	1	133	
1	53	1	9254	1	42	110	
1	1	1	9085	1	1	149	
1	1	1	9773	1	1	127	
1	18	1	9238	1	1	137	
1	17	2	9500	1	1	135	
1	1	1	9130	1	1	137	
1	1	1	9853	1	1	140	

Raw Content

```
data:text/csv;base64,RmFrZV9TdHVkZW50SUQsTWFyaXRhbCBzdGF0dXMsQXBwbGljYXRpb24gbW9kZSxBcHBsaWNhdGlvbiBvcmRlcixDb3Vyc2UsRGF5dGltZS9ldmVuaW5nIGF0dGVuZGFuY2UsUHJldmlvdXMgc
XVhbGlmaWNhdGlvbixQcmV2aW91cyBxdWF...
```

Image 3: Data uploaded in a table

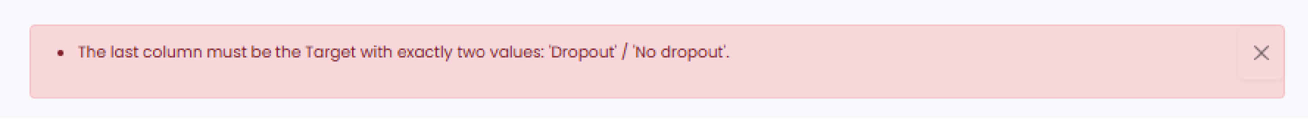
- **Error Handling**: If a dataset fails to meet any of the requirements, the tool displays an **error message (image 5)** indicating the problem and highlights the unmet item on the **checklist in red (image 4).**



Requirements check

- ☑ File is .csv
- ☑ At least one continuous (float) column
- ☒ Target in the last column with two values: 'Dropout' / 'No dropout'

Image 4: checklist with fail



- The last column must be the Target with exactly two values: 'Dropout' / 'No dropout'.

Image 5: Wrong file error message

Once the correct dataset has been loaded, the user must press the “start” button for the tool to proceed to display the dashboard hub.

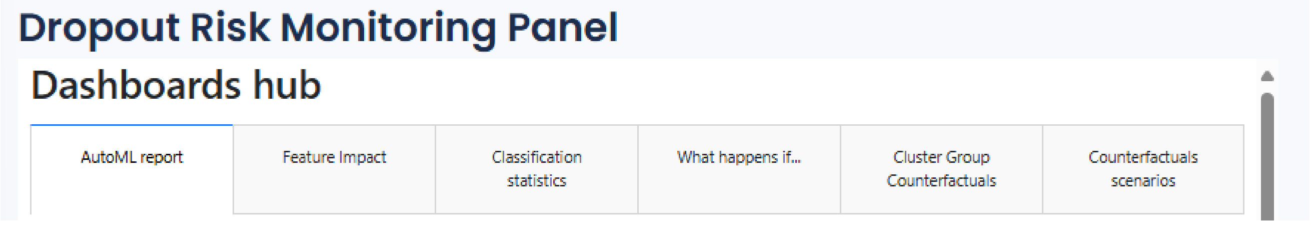

Dropout Risk Monitoring Panel
Dashboards hub
AutoML report
Feature Impact
Classification statistics
What happens if...
Cluster Group Counterfactuals
Counterfactuals scenarios

Image 6: Dashboard’s hub

3.2. Dashboard hub loading screen

When running the tool, the user accesses the Hub's main screen, which is the entry point for all the system's functionalities.

This screen is the entry point for all the system's functionalities. Users can navigate the analysis modules, which include:

- **AutoML report tab:** This section includes different blocks of information represented by boards, Performance graphs, and correlations.
- **Feature impact tab:** Which includes graphs about the feature impact on the model
- **Classification statistics tab:** With elements that indicate performance metrics and classification plots
- **What Happens if analysis tab:** This tab is organized into several functional blocks: selecting a student, contributions of each feature to the prediction plot, result of the prediction for the student and the possibility of adjusting the values of the features to see the impact on the results.
- **Cluster Group Counterfactuals tab:** There are two options: automatic and manual cluster creation, select the cluster to analyze and see the results in tables and plots.
- **Individual counterfactuals scenarios tab:** In this tab the user can decide the number of scenarios to calculate and see the results in tables and plots.

The dashboard's purpose is to allow intuitive visualization of predictions and risk factors