# 

**User Manual**

**For Client Company’s Management Staff**

**20 November 2024**

**Team Members:**

Kavyanjali 2203390

Koh Chun Ince 2200960

Caleb Lee Jia Le 2201250

Muhammad Muslinmin Bin Mohd Saleh 2201360

Reuben Low Yu Xiang 2200931

Stafford Ho Sheng Xian 2201218

Yugendren S/o Sooriya Moorthi 2201512

[1 Minimum System Requirements 3](#_uz7i0jof2pej)

[2 Directory Structure Overview 3](#_kbzr2jxkufkt)

[3 Installation Guide 3](#_g9npwxtaug0i)

[4 Step-by-Step Usage Guide 3](#_nm80hddt6xwh)

[4.1 Using the CLI 4](#_8kvzy0o4g181)

[4.1 Using the GUI 5](#_mf7g5ji8c3vg)

[4 Troubleshooting 6](#_ayhg0puizajm)

## **1 Minimum System Requirements**

**Hardware**:

* CPU: Intel i3/i5 or equivalent.
* Memory: 8 GB recommended.
* Storage: 100 GB HDD/SSD.

**Software**:

* OS: Windows 10/11
* Bash shell terminal, Windows Command Prompt for running scripts.
* Jupyter notebook

## **2 Directory Structure Overview**

The program has the following file structure:

Team5-MS\_2/

├── Win\_10-Python\_v3.11/

│ ├── EDA\_DATA

│ ├── EXECUTABLES

│ ├── ML\_DATA

│ ├── myenv

│ ├── OTH\_DATA

│ ├── PURE\_SRC\_CODE

│ ├── SCRIPTS\_CFG

│ ├── requirements.txt

│ ├── README-first.md

├── JupyterNB-Python\_v3.11/

│ ├── EDA\_DATA

│ ├── EXECUTABLES

│ ├── ML\_DATA

│ ├── myenv

│ ├── OTH\_DATA

│ ├── PURE\_SRC\_CODE

│ ├── SCRIPTS\_CFG

│ ├── COMBINED\_PIPELINE\_DEMO

│ ├── EDA\_PIPELINE\_DEMO

│ ├── ML\_PIPELINE\_DEMO

│ ├── requirements.txt

│ ├── README-first.md

├── DOCS

├── .git

Where both folders Win\_10-Python\_v3.11 and JupyterNB-Python\_v3.11 contains:

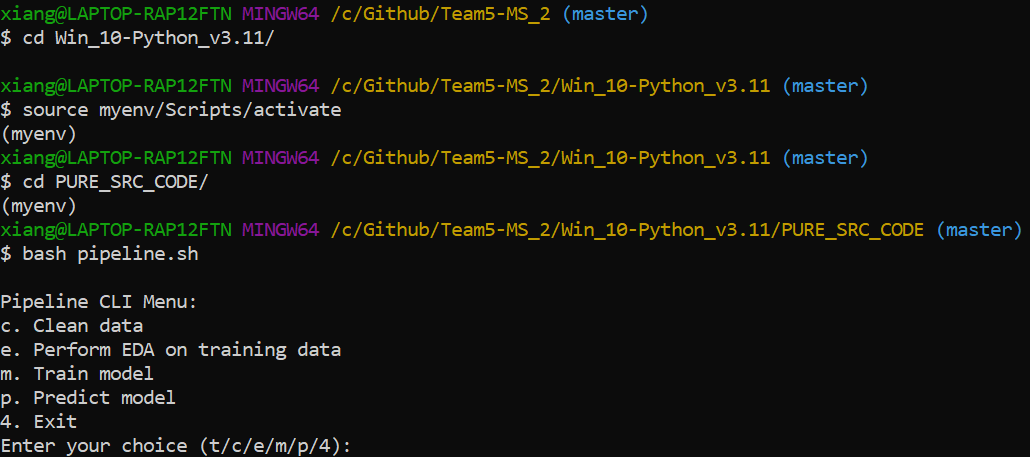
* **EDA\_DATA**: Location of outputs from the exploratory data analysis e.g. graphs.
* **EXECUTABLES**: Location of binary executable files of the program.
* **ML\_DATA**: Location of folders:
  + model\_outputs: Contains the machine learning models trained on the train dataset.
  + visualize\_outputs: Contains the evaluation metrics and visualization of the performed predictions on the test datasets.
* **myenv**: Location of the virtual environment.
* **OTH\_DATA**: Location of folders:
  + training\_data: Raw datasets that have not been cleaned.
  + cleaned\_data: Cleaned datasets that are processed from the raw datasets.
* **PURE\_SRC\_CODE**: Location of pipeline.sh bash file to run pipeline as well as underlying python scripts for initiating the pipeline.
* **SCRIPTS\_CFG**: Contains the configuration text file ‘config.txt’ that allows the user to specify the folder paths, file name suffix and the model training parameters.
* **README-First.txt**: Contains instructions on installing the necessary packages and to run the program that are specific to either running the program using the python pipeline or the jupyter notebook pipeline.
* **Requirements.txt**: Contains the necessary packages needed to be installed to run the program that is executed by ‘pip install -r requirements.txt’.

The JupyterNB-Python\_v3.11 folder contains an additional three folders to run the Jupyter notebook pipeline:

* **EDA\_PIPELINE\_DEMO**: Contains the notebooks clean\_data, perform\_eda needed to execute exploratory data analysis on datasets.
* **ML\_PIPELINE\_DEMO**: Contains the notebooks clean\_data, model\_training and perform\_prediction needed to train the model on a cleaned dataset and to perform prediction on test dataset to generate evaluation metrics on the models’ performance.
* **COMBINED\_PIPELINE\_DEMO**: Contains the entire pipeline that consists of EDA\_PIPELINE\_DEMO and ML\_PIPELINE\_DEMO.

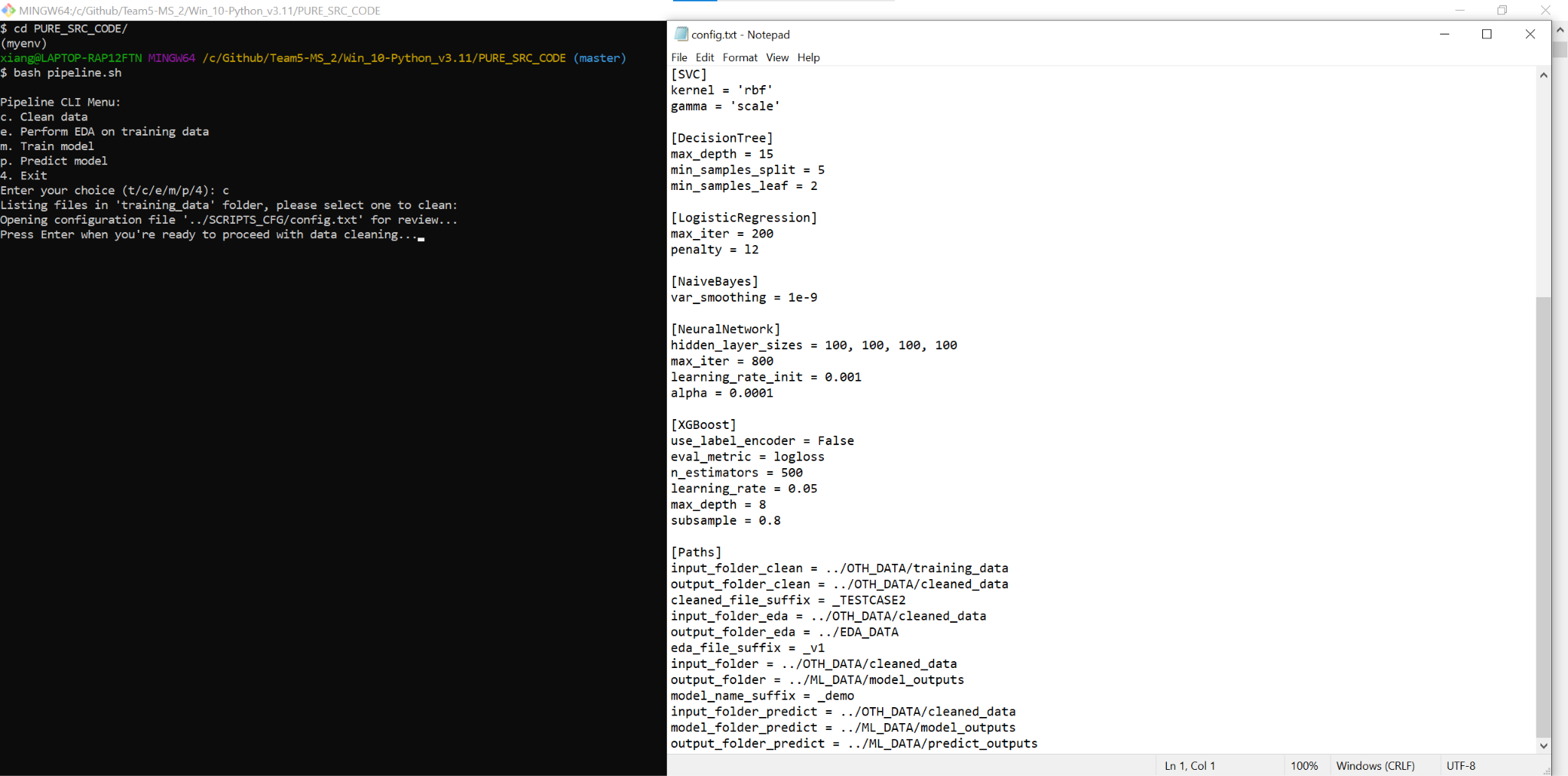
## **3 Step-by-Step Usage Guide: Python script pipeline**

Step 1: Executing the bash file

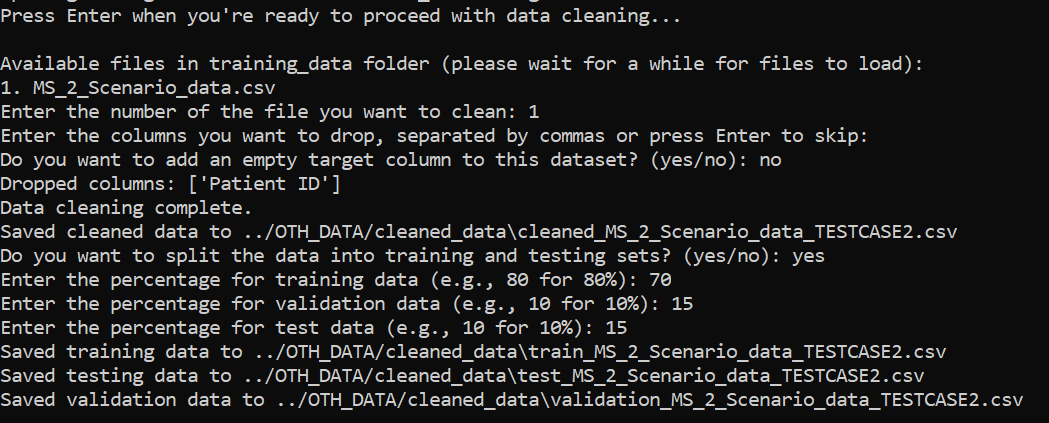


* Navigate to the Win\_10-Python\_v3.11 folder
* Start the virtual environment by entering the command:
  + source myenv/Scripts/activate
* Navigate to the PURE\_SRC\_CODE folder by entering the command:
  + cd PURE\_SRC\_CODE
* To start the python script pipeline, enter the command:
  + bash pipeline.sh

Step 2: Perform cleaning data on datasets:



* Once the pipeline CLI menu is opened, enter choice ‘c’.
* A configuration text file will be opened, scroll down to the paths section and modify the suffix of the clean dataset output under cleaned\_file\_suffix = \_example where \_example is your desired suffix.
* If you wish, you may modify the default folder path under input\_folder\_clean = ../OTH\_DATA/training\_data and output\_folder\_clean = ../OTH\_DATA/cleaned\_data.
* Once you are satisfied with the name suffix and folder paths, close the configuration text file and press the enter key on the terminal windows to move onto the next step.



* The available raw datasets will be displayed.
* Enter the file number of your choice corresponding to the list of available files, in this case we choose file number ‘1’.
* You will be asked what columns in the raw dataset you want to remove, continue pressing the enter key to proceed.
* You will be asked whether you want to insert an empty target column if your raw test dataset does not have any, continue pressing the enter key to proceed if you do not want to insert the target column. *(Note: Our prediction functionality is able to handle test datasets that do not have target columns.)*
* You will be asked if you want to split the datasets into train/test/validation splits of train/test splits.
  + For example, suppose you want a split of 70% train, 15% test, 15% validation set, you enter:
    - Enter the percentage for training data (e.g., 80 for 80%): 70
    - Enter the percentage for validation data (e.g., 10 for 10%): 15
    - Enter the percentage for test data (e.g., 10 for 10%): 15
  + For example, suppose you want a split of 80% train, 20% test, 0% validation set, you enter:
    - Enter the percentage for training data (e.g., 80 for 80%): 80
    - Enter the percentage for validation data (e.g., 10 for 10%): 0
    - Enter the percentage for test data (e.g., 10 for 10%): 20
* Once you have entered the parameters, the raw dataset will be cleaned and split according to you choice.

Step 3: Perform Exploratory Data Analysis (EDA) (e)

* Select the EDA Option
  + Enter e to perform EDA.
* Choose Input File
  + Select a .csv file from the cleaned data folder.
* EDA Outputs
  + Generates visualizations like box plots, scatter plots, and heatmaps.
  + Saves plots and summaries in the output folder (e.g., ../EDA\_DATA).

Step 4: Train Model (m)

* Select the Model Training Option
  + Enter m to train models.
* Choose Input File
  + Select a .csv file from the cleaned data folder.
* Select Features
  + Choose columns to use as features for model training.
* Specify Target Column
  + Provide the column name containing the target variable (e.g., Obesity\_Level).
* Choose Models
  + Select one or more models from the list (e.g., RandomForest, XGBoost).
  + Type all to train all models.
* Output
  + Trained models are saved as .pkl files in the specified output folder.

Step 5: Predict Model (p)

* Select the Prediction Option
  + Enter p to make predictions.
* Choose Input File
  + Select a .csv file containing test data.
* Load Model
  + Select a pre-trained model from the output folder.
* View Predictions
  + Predictions are saved as formatted .xlsx files.
  + Additional metrics (e.g., accuracy, confusion matrix) are saved in the output folder.

Step 6: Exit (4)

* Enter 4 to exit the pipeline.

View Outputs:

* Outputs are saved in designated folders based on the pipeline stage:
  + Cleaned data: OTH\_DATA/cleaned\_data.
  + EDA results: OTH\_DATA/EDA\_DATA.
  + Trained models: ML\_DATA/model\_outputs.
  + Predictions: ML\_DATA/predict\_outputs.

**Notes**

* Configuration File: All paths and settings are controlled via the configuration file.
* Output Directories: Ensure output directories specified in the config file exist before running.

## **4 Step-by-Step Usage Guide: Jupyter notebook**

Step 1: Start Jupyter Notebook

* Open Terminal
* Launch a terminal or command prompt on your machine.
* Navigate to the Folder
* Use the cd command to navigate to the directory containing the pipeline notebooks by entering the command from the program home directory:
  + cd JupyterNB-Python\_v3.11
  + cd COMBINED\_PIPELINE\_DEMO
* Run the following command:
  + jupyter notebook
* This will open the Jupyter Notebook interface in your default web browser.

Step 2: Clean Data

* Open the Clean Data Notebook
  + In the Jupyter Notebook interface, locate and open the clean\_data.ipynb notebook.
* Load Configuration File
  + The notebook may require a configuration file to specify input and output directories. Provide the path as instructed in the notebook.
* Select Input File
  + Use the provided dropdown or file selection cell to choose a .csv file for cleaning.
* Perform Cleaning
  + Follow the prompts in the notebook to specify columns to drop, add a target column if necessary, and handle missing values.
  + Optionally, calculate derived metrics like BMI.
* Save Output
  + The cleaned data will be saved to the output folder as specified in the notebook.

Step 3: Perform Exploratory Data Analysis (EDA)

* Open the EDA Notebook
  + In the Jupyter Notebook interface, open the perform\_eda.ipynb notebook.
* Load Cleaned Data
  + Select a cleaned .csv file as the input for EDA.
* Generate Visualizations
  + Run the cells to produce plots such as:
    - Distribution plots
    - Correlation heatmaps
    - Boxplots
    - Scatter plots
* Save Results
  + The notebook will save all plots and summaries to the specified EDA output folder.

Step 4: Train Model

* Open the Model Training Notebook
  + Locate and open the model\_training.ipynb notebook.
* Load Cleaned Data
  + Choose a .csv file containing cleaned data for training.
* Specify Features and Target Column
  + Select the features to use for training and the target column.
* Choose Models
  + The notebook supports training models like RandomForest, XGBoost, and others. Follow prompts to select models for training.
* Save Trained Models
  + Trained models will be saved as .pkl files in the specified output folder.

Step 5: Perform Prediction

* Open the Prediction Notebook
  + Open the perform\_prediction.ipynb notebook.
* Load Test Data
  + Select a .csv file containing test data.
* Load Pre-trained Models
  + Choose a trained model file (.pkl) from the output folder.
* Make Predictions
  + The notebook will generate predictions and evaluate metrics such as accuracy and confusion matrix.
* Save Results
  + Predictions and evaluation metrics are saved as .xlsx and .png files in the output folder.

**Notes**

* Configuration File: Ensure the configuration file is updated with correct paths for input and output directories.
* Output Directories: Verify that all necessary output directories exist before running the notebooks.

## **5 Troubleshooting for python pipeline**

## **Bash Script Fails to Execute**

· Ensure the script has execution permissions:  
 `chmod +x pipeline.sh`

· Check if the bash script is in the correct directory and is being executed from the right path.

· Verify that Bash is installed on your system by running:  
 `bash --version`

## **Missing Dependencies**

· Install the required dependencies:  
 `pip install -r requirements.txt`

· If specific libraries are missing, install them individually:  
 `pip install pandas scikit-learn matplotlib seaborn xgboost`

## **File Not Found Errors**

· Confirm that the specified file paths in the configuration file are correct.

· Ensure input files are in the correct directory (e.g., `../OTH\_DATA/cleaned\_data`).

· If the configuration file is missing, recreate it with the necessary settings.

## **Output Files Not Generated**

· Ensure the output directories specified in the configuration file exist.

· Verify that the Python scripts executed correctly without errors.

· Check if the paths in the configuration file match the desired output location.

## **Script Crashes During Execution**

· Review the error messages in the terminal to identify the cause.

· Ensure that the input files are not corrupted and conform to expected formats.

· Test with smaller datasets if memory issues are suspected.

## **Trained Models Not Saving**

· Ensure the `joblib` library is installed:  
 `pip install joblib`

· Verify that the output folder for models is correctly specified in the configuration file.

## **Prediction Errors**

· Ensure the test data contains the same features used during training.

· If the target column is missing, proceed without it but note that evaluation metrics like accuracy will not be calculated.

## **Permission Issues**

· Check directory permissions and ensure you have write access.

· Run the Python scripts with elevated privileges (administrator mode) if necessary.

## **Version Conflicts**

· Check installed library versions:  
 `pip list`

· Update libraries to compatible versions using:  
 `pip install --upgrade <library\_name>`

· Use a virtual environment to manage dependencies:  
 `python -m venv env`

## **Slow Execution**

· Optimize data by reducing the dataset size for initial tests.

· Use a more powerful system or cloud resources for larger datasets.

· Profile your scripts to identify slow sections using:  
 `time` or `cProfile`.

## **6 Troubleshooting for jupyter notebook**

## **Jupyter Notebook Fails to Launch**

· Ensure Jupyter is installed:  
 `pip install notebook`

· Check if Jupyter is added to your PATH. If not, activate the virtual environment or use:  
 `python -m notebook`

· Verify your Python installation is functional:  
 `python --version`

## **Missing Dependencies**

· Install the required dependencies:  
 `pip install -r requirements.txt`

· If specific libraries are missing, install them individually:  
 `pip install pandas scikit-learn matplotlib seaborn xgboost`

## **File Not Found Errors**

· Confirm that the specified file paths in the configuration file are correct.

· Ensure input files are in the correct directory (e.g., `../OTH\_DATA/cleaned\_data`).

· If the configuration file is missing, recreate it with the necessary settings.

## **Output Files Not Generated**

· Ensure the output directories specified in the configuration file exist.

· Verify that the notebook cells for saving outputs have been executed without errors.

· Check if the paths in the configuration file match the desired output location.

## **Kernel Crashes or Freezes**

· Restart the kernel via the Jupyter interface (`Kernel > Restart`).

· Free up system resources by closing unused applications.

· Use smaller datasets for testing to reduce memory usage.

## **Visualization Issues**

· Ensure `matplotlib` and `seaborn` are installed:  
 `pip install matplotlib seaborn`

· Check the output directory for saved plot files.

· If the plot resolution is low, adjust the DPI in the notebook cells that generate plots:  
 `plt.savefig("output.png", dpi=300)`

## **Trained Models Not Saving**

· Ensure the `joblib` library is installed:  
 `pip install joblib`

· Verify that the output folder for models is correctly specified in the configuration file.

## **Prediction Errors**

· Ensure the test data contains the same features used during training.

· If the target column is missing, proceed without it but note that evaluation metrics like accuracy will not be calculated.

## **Permission Issues**

· Check directory permissions and ensure you have write access.

· Run Jupyter Notebook with elevated privileges (administrator mode) if necessary.

## **Version Conflicts**

· Check installed library versions:  
 `pip list`

· Update libraries to compatible versions using:  
 `pip install --upgrade <library\_name>`

· Use a virtual environment to manage dependencies:  
 `python -m venv env`

## **Slow Execution**

· Optimize data by reducing the dataset size for initial tests.

· Use a more powerful system or cloud resources for larger datasets.

· Profile your notebook to identify slow sections using:  
 `%timeit`