

|  |
| --- |
| Adil Alami  7-8-2024 |

|  |
| --- |
| Faculté Polydisciplinaire – Taroudant |
| **Compte Rendu : ML I – Mini Project** |
| Master – Big Data et Intelligence Artificielle (BDIA)  Département : GI  Option : BDIA (S2)  A.U : 2023 - 2024 |

Summary

[Summary 1](#_Toc171355737)

[Illustrations List 3](#_Toc171355738)

[Introduction 1](#_Toc171355739)

[Chapter I. Methodology 2](#_Toc171355740)

[I. Motivation 3](#_Toc171355741)

[II. Objectives 3](#_Toc171355742)

[III. Methodology 3](#_Toc171355743)

[IV. Implementation 4](#_Toc171355744)

[Chapter II. Implementation 5](#_Toc171355745)

[I. Design aspect 6](#_Toc171355746)

[1. Classification task 6](#_Toc171355747)

[2. Clustering task 7](#_Toc171355748)

[II. Technologies used 7](#_Toc171355749)

[1. Flask for Backend Development 7](#_Toc171355750)

[2. Frontend Development with HTML, CSS, and JavaScript 8](#_Toc171355751)

[3. Machine Learning and Data Analysis with Python 8](#_Toc171355752)

[III. Datasets used 9](#_Toc171355753)

[1. Classification dataset 9](#_Toc171355754)

[2. Clustering dataset 10](#_Toc171355755)

[IV. Algorithms used 11](#_Toc171355756)

[1. Classification 11](#_Toc171355757)

[a. Use of Pretrained EfficientNet-B0 for Image Feature Extraction 11](#_Toc171355758)

[b. Multi-Class Logistic Regression 12](#_Toc171355759)

[V. Result 13](#_Toc171355760)

[2. Engagements.csv 13](#_Toc171355761)

[3. Eye Aspect Ration (for tiredness detection) csv 13](#_Toc171355762)

[4. Face counts (to detect presence and number of individuals) csv 14](#_Toc171355763)

[5. Engagement chart (real time) 14](#_Toc171355764)

[6. Eye Aspect ratio chart (real time) 15](#_Toc171355765)

[7. Face Count Chart (real time) 15](#_Toc171355766)

[8. Clustering group (manual input) 16](#_Toc171355767)

[VI. Conclusion 16](#_Toc171355768)

Illustrations List

[Figure ‎I‑1 : Classification diagram 4](#_Toc171355561)

[Figure ‎I‑2 : Clustering diagram 4](#_Toc171355562)

[Figure ‎II‑1 : Calssification page 6](#_Toc171355563)

[Figure ‎II‑2 : Classification page 6](#_Toc171355564)

[Figure ‎II‑3 : Clustering page 7](#_Toc171355565)

[Figure ‎II‑4 : Flask Logo 7](#_Toc171355566)

[Figure ‎II‑5 : HTML, CSS, JS Logos 8](#_Toc171355567)

[Figure ‎II‑6 : NumPy, Pandas, Scikit-Lear, PyTorch stack Logos 9](#_Toc171355568)

[Figure ‎II‑7 : Classification data 9](#_Toc171355569)

[Figure ‎II‑8 : Clustering dataset 10](#_Toc171355570)

[Figure ‎II‑9 : Comparing deep learning architectures 11](#_Toc171355571)

[Figure ‎II‑10 : engagement.csv 13](#_Toc171355572)

[Figure ‎II‑11 : ear\_values.csv 13](#_Toc171355573)

[Figure ‎II‑12 : face\_count.csv 14](#_Toc171355574)

[Figure ‎II‑13 : engagement chart (real time) 14](#_Toc171355575)

[Figure ‎II‑14 : Tiredness chart (real time) 15](#_Toc171355576)

[Figure ‎II‑15 : Faces count (real time) 15](#_Toc171355577)

[Figure ‎II‑16 : 3D visualization of the student cluster group 16](#_Toc171355578)

Introduction

In the evolving landscape of educational technology, understanding and analyzing student engagement has become crucial for enhancing learning experiences and outcomes. This project leverages Big Data and Artificial Intelligence techniques to monitor and analyze student engagement in real-time using a web-based application. The primary objective is to create a dynamic and interactive system that collects data on student activities, performs clustering analysis to identify patterns in engagement, and visualizes these insights through intuitive graphs and charts.

# Methodology

1. Motivation

Effective engagement is a key indicator of student success. Traditional methods of monitoring engagement, such as manual observations and surveys, are often time-consuming and subjective. By integrating advanced data analytics and machine learning, this project aims to provide a scalable and objective solution for tracking and improving student engagement.

1. Objectives
2. **Real-Time Data Collection:** Implement a system to collect real-time data on student activities, including facial expressions and eye aspect ratio (EAR), using computer vision techniques.
3. **Engagement Analysis:** Use clustering algorithms to classify student engagement levels based on the collected data.
4. **Interactive Visualization:** Develop an interactive web interface to visualize engagement metrics, cluster analysis results, and predicted engagement labels.
5. **Actionable Insights:** Provide educators with actionable insights to tailor instructional strategies and enhance student engagement.
6. Methodology
7. **Data Collection:** The system collects data on student engagement through a webcam and analyzes it using computer vision techniques to extract features such as face count and EAR values.
8. **Data Processing:** The collected data is processed and sent to a Flask backend, where the frames are classified into a behavior based on an already trained machine learning algorithm. And for the clustering part clustering analysis is performed using the K-Means algorithm.
9. **Visualization:** The results of the classification and clustering analysis, including engagement levels and cluster centers, are visualized using Plotly.js to provide real-time feedback to educators and students.
10. **Machine Learning:** The K-Means clustering algorithm is used to identify patterns in student engagement, and the results are continuously updated as new data is collected. The real time classification using both deep learning for features extraction and machine learning for classification insures continuous feedback about the students behavior.

A close-up of a document

Description automatically generated

Figure ‎I‑1 : Classification diagram

A close-up of a piece of paper

Description automatically generated

Figure ‎I‑2 : Clustering diagram

1. Implementation

* **Frontend:** A native HTML, CSS, and JS application serves as the user interface, providing real-time visualizations and interaction capabilities.
* **Backend:** A Flask server handles data processing, clustering analysis, and serves the processed data to the frontend.
* **Visualization:** Plotly.js is used for dynamic and interactive charting, allowing users to explore engagement metrics in 2D and 3D scatter plots and bar charts.

# Implementation

1. Design aspect

A simple and intuitive design was used for this project, to insure seamless user experience.

## Classification task

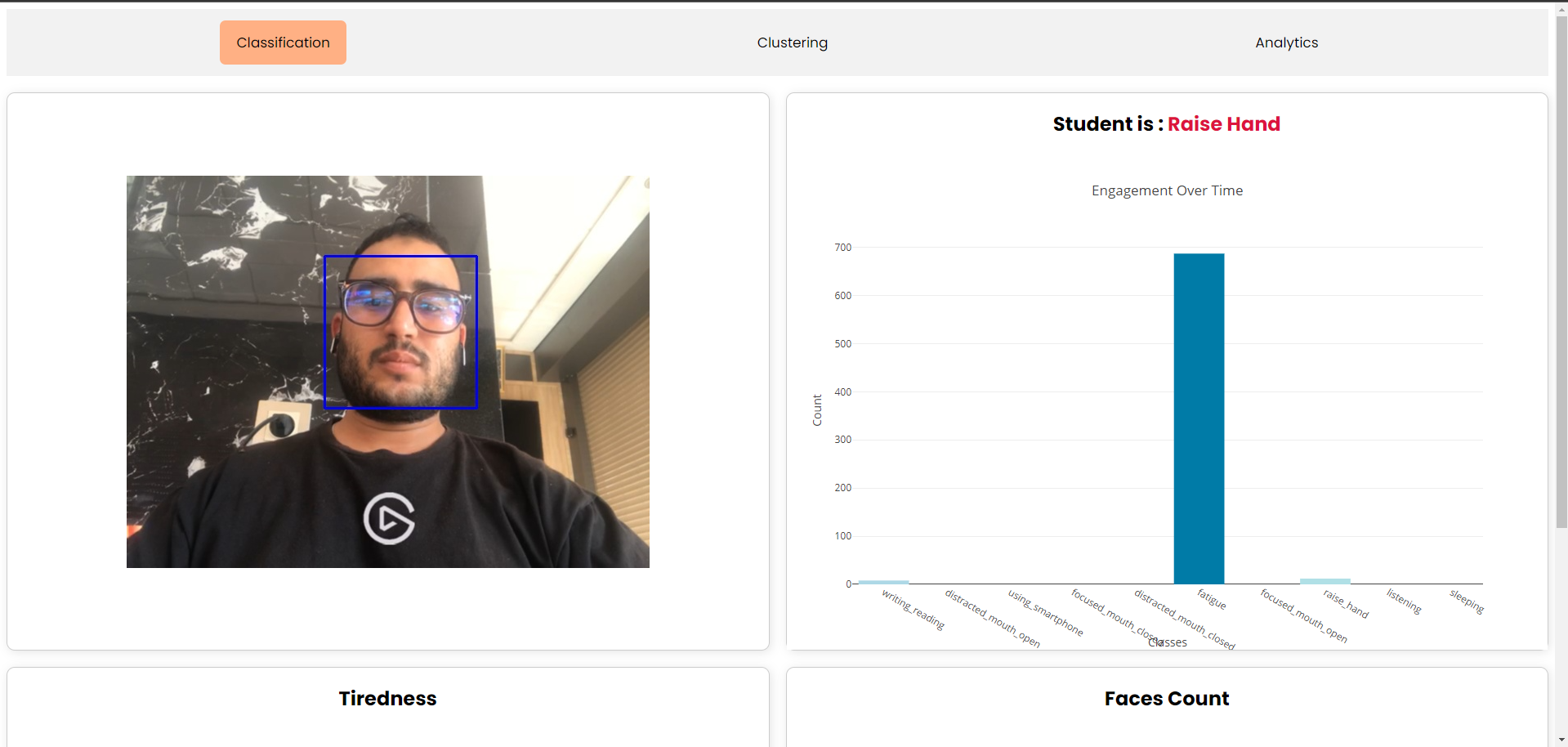


Figure ‎II‑1 : Calssification page

A screenshot of a computer

Description automatically generated

Figure ‎II‑2 : Classification page

## Clustering task

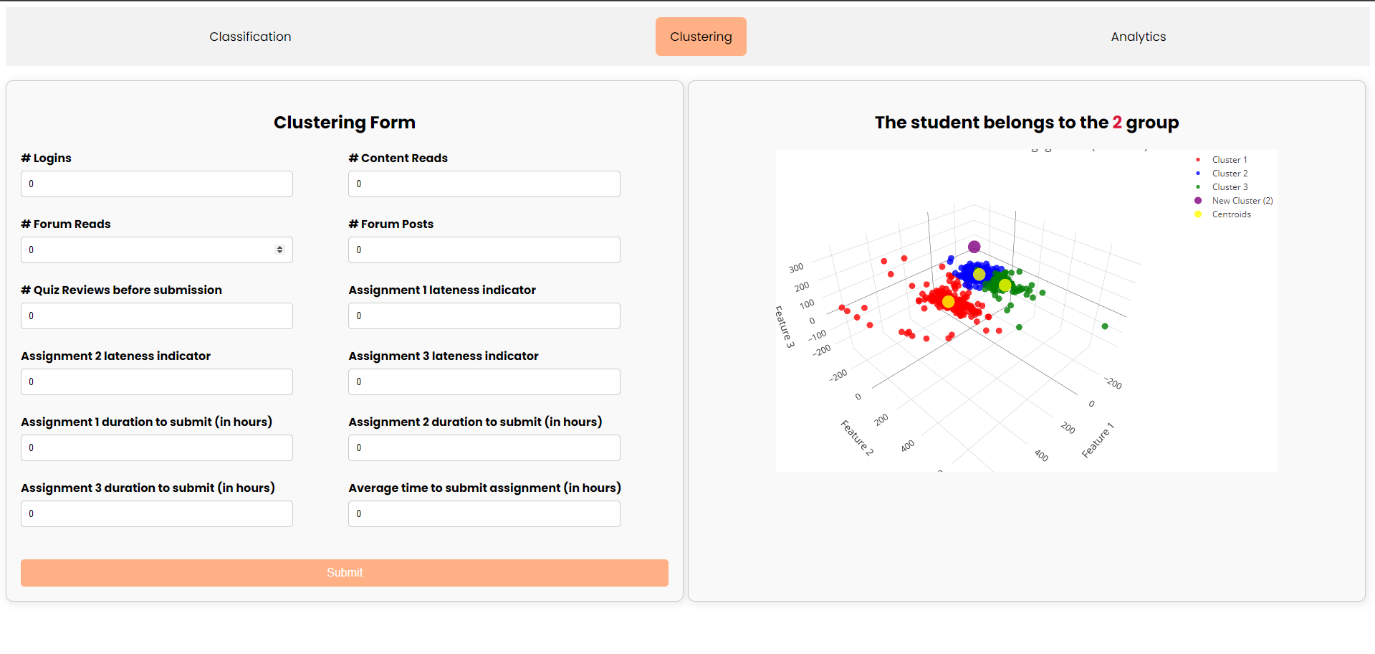


Figure ‎II‑3 : Clustering page

1. Technologies used

In this project, a variety of technologies have been employed to achieve the goal of monitoring and analyzing student engagement in real-time. The choice of technologies spans both the backend and frontend development, as well as the implementation of machine learning and data visualization.

## Flask for Backend Development



Figure ‎II‑4 : Flask Logo

Flask is a lightweight and versatile web framework for Python, which was chosen for the backend development of this project. Flask provides the necessary tools and libraries to build a robust backend capable of handling data processing, communication with the frontend, and performing machine learning tasks. The main roles of Flask in this project include:

* **Data Processing:** Handling incoming data from the frontend, processing it for machine learning analysis.
* **API Endpoints:** Serving as an interface between the frontend and the machine learning models, providing endpoints for data submission and retrieval.
* **Integration:** Coordinating the flow of data between different components of the application.

## Frontend Development with HTML, CSS, and JavaScript

The frontend of the application is developed using a combination of HTML, CSS, and JavaScript, providing a responsive and user-friendly interface. These technologies enable the creation of a dynamic and interactive web application that users can easily navigate. Key aspects include:

* **HTML:** Structuring the content of the web application.
* **CSS:** Styling the application to enhance the user experience with an intuitive layout and design.
* **JavaScript:** Adding interactivity and handling dynamic content updates, particularly in conjunction with Plotly.js for data visualization.



Figure ‎II‑5 : HTML, CSS, JS Logos

## Machine Learning and Data Analysis with Python

Python is the core language used for machine learning and data analysis in this project. Leveraging Python's extensive libraries, the project implements advanced analytics to classify and predict student engagement. Key Python libraries and tools include:

* **NumPy and Pandas:** For data manipulation and preprocessing.
* **Scikit-learn:** For implementing the K-Means clustering algorithm used to analyze and categorize student engagement data and the Logistic Regression classification algorithm used to classify students’ behavior.
* **PyTorch:** For building and training deep learning for feature extraction.

|  |  |
| --- | --- |
| A blue and black logo  Description automatically generated | A blue and black text  Description automatically generated |
| A close-up of a logo  Description automatically generated | Pytorch logo - Social media & Logos Icons |

Figure ‎II‑6 : NumPy, Pandas, Scikit-Lear, PyTorch stack Logos

1. Datasets used

## Classification dataset

We will use frames taken from the video stream to detect the face (faces count), tiredness, and predict the engagement class, for we used a publicly available dataset on kaggle.

A collage of a person and person

Description automatically generated

Figure ‎II‑7 : Classification data

## Clustering dataset

One of the datasets I found on github (link), provided by Dr. MohammadNoor Injadat, Dr. Abdallah Moubayed; and Dr. Abdallah Shami under the MIT license that they used in their published paper entitled: Relationship Between Student Engagement and Performance in E-Learning Environment Using Association Rules. The collected dataset is from a second year undergraduate Science course offered at a North American University that was given in a blended format (i.e. course had both a face-to-face component and online component). The dataset contains features such as:

* Student Id
* Number of Logins
* Number of Content Reads
* Number of Forum Reads
* Number of Forum Posts
* Number of Quiz Reviews
* Assignment 1 lateness indicator
* Assignment 2 lateness indicator
* Assignment 3 lateness indicator
* Assignment 1 duration to submit (in hours)
* Assignment 2 duration to submit (in hours)
* Assignment 3 duration to submit (in hours)
* Average Assignment duration to submit (in hours)
* Engagement Level

A screenshot of a computer screen

Description automatically generated

Figure ‎II‑8 : Clustering dataset

1. Algorithms used

## Classification

### Use of Pretrained EfficientNet-B0 for Image Feature Extraction

A critical component of this project is the use of advanced deep learning models for image feature extraction. Specifically, the project employs a pretrained EfficientNet-B0 model to process and extract meaningful features from student images. This section delves into the role and advantages of using EfficientNet-B0 in the context of this project.

EfficientNet-B0 is a state-of-the-art convolutional neural network (CNN) architecture that balances model efficiency and performance. Developed by Google, EfficientNet introduces a novel scaling method that uniformly scales network width, depth, and resolution using a compound coefficient. This approach leads to significant improvements in model accuracy and computational efficiency compared to traditional CNN architectures.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Parameters | Top-1 Accuracy | Top-5 Accuracy | Model Size (MB) | Year |
| AlexNet | 62.3M | 56.5% | 79.1% | 240 | 2012 |
| VGG16 | 138M | 71.5% | 89.8% | 528 | 2014 |
| Inception v3 | 23.8M | 77.9% | 93.7% | 92 | 2015 |
| ResNet-50 | 25.6M | 76.0% | 92.9% | 98 | 2015 |
| DenseNet-121 | 8M | 74.9% | 92.3% | 32 | 2016 |
| EfficientNet-B0 | 5.3M | 77.1% | 93.3% | 20 | 2019 |

Figure ‎II‑9 : Comparing deep learning architectures

Based on the comparative table, EfficientNet-B0 proves to be an excellent choice for this project due to its balance of high accuracy, low parameter count, and compact model size. It achieves a top-1 accuracy of 77.1% and a top-5 accuracy of 93.3% with only 5.3 million parameters, significantly outperforming older architectures like AlexNet and VGG16 in terms of efficiency and performance. The smaller model size (20 MB) also facilitates faster loading and deployment, making EfficientNet-B0 ideal for applications with limited computational resources. This combination of attributes makes EfficientNet-B0 a superior choice for efficient and accurate feature extraction in image-based tasks.

### Multi-Class Logistic Regression

Logistic regression is not limited to binary classification problems; it can also be extended to handle multi-class classification tasks. Multi-class logistic regression models the probabilities that a given input belongs to one of several classes. It uses the softmax function to transform the outputs of a linear regression model into probabilities for each class, which are then used to make predictions.

In multi-class logistic regression, the softmax function is applied to the linear combinations of the input features and their corresponding weights for each class. The output is a set of probabilities, each representing the likelihood of the input belonging to one of the classes.

Given a set of input features and corresponding weights and bias terms for each class , the linear combination for class is computed as:

where:

is the weight vector for class

is the input feature vector

is the bias term for class

The output of the multi-class logistic regression model, representing the probability of the input belonging to class , is calculated using the softmax function:

where:

is the linear combination for class

is the number of classes

is the base of the natural logarithm

The softmax function ensures that the output probabilities sum to 1, providing a valid probability distribution across all classes.

Prediction

To predict the class label, the class with the highest probability is selected:

In other words, the input is classified as belonging to the class with the highest predicted probability.

1. Result

The classification results in 3 csv files for further analysis and visualization.

## Engagements.csv

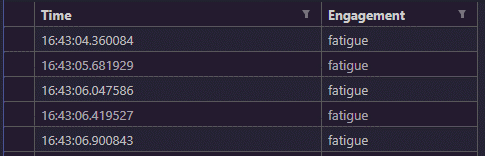


Figure ‎II‑10 : engagement.csv

## Eye Aspect Ration (for tiredness detection) csv

A screenshot of a computer

Description automatically generated

Figure ‎II‑11 : ear\_values.csv

## Face counts (to detect presence and number of individuals) csv

A screenshot of a computer

Description automatically generated

Figure ‎II‑12 : face\_count.csv

## Engagement chart (real time)

A graph with blue bars

Description automatically generated

Figure ‎II‑13 : engagement chart (real time)

## Eye Aspect ratio chart (real time)

A graph with red lines

Description automatically generated

Figure ‎II‑14 : Tiredness chart (real time)

## Face Count Chart (real time)

A graph with blue lines and numbers

Description automatically generated

Figure ‎II‑15 : Faces count (real time)

## Clustering group (manual input)

A screen shot of a graph

Description automatically generated

Figure ‎II‑16 : 3D visualization of the student cluster group

1. Conclusion

In conclusion, this project successfully demonstrates the integration of Big Data and AI techniques to monitor and analyze student engagement in real-time. Through the development of a web-based application, we have created a system that not only collects and processes data but also provides meaningful visualizations and insights. The use of clustering algorithms and machine learning models has enabled the identification of patterns in student behavior, offering educators a valuable tool for enhancing teaching strategies and improving student outcomes. The findings underscore the potential of data-driven approaches in education and pave the way for future research and development in this field. The project highlights the importance of leveraging technology to foster an engaging and effective learning environment, ultimately contributing to the advancement of educational practices.