ONLINE STUDENT ACTIVENESS SYSTEM

Minor project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Abhishek Sharma(191258)

Shivam Singh Negi(191260)

UNDER THE SUPERVISION OF

Mr.Arvind Kumar

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, 173234, Himachal Pradesh, INDIA

DECLARATION

I hereby declare that this project has been done by me under the supervision of **Dr.Arvind Kumar,Assistant Professor (Grade-2),** Jaypee University of Information Technology. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:
Dr.Arvind Kumar,
Assistant Professor (Grade-2),
Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology

Submitted by: Abhishek Sharma(191258)
Shivam Singh Negi(191260)
Computer Science & Engineering Department,
Jaypee University of Information Technology

CERTIFICATE

This is to certify that the work which is being presented in the project report titled **“Online Student Activeness System”** in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by “**Abhishek Sharma(191258),Shivam Singh Negi(191260)**” during the period from January 2022 to May 2022 under the supervision of Dr. Arvind Kumar, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

Abhishek Sharma(191258)
Shivam Singh Negi(191260)

The above statement made is correct to the best of my knowledge.

Dr. Arvind Kumar
Assistant Professor(Grade-2)
Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat,

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible to complete the project work successfully.

I am really grateful and wish my profound indebtedness to Supervisor **Dr./Mr./Mrs. Arvind Kumar, Designation**, Department of CSE Jaypee University of Information Technology,Wakhnaghat. Deep Knowledge & keen interest of my supervisor in the field of "**Machine Learning**" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to **Dr./Mr./Mrs. Arvind Kumar,** Department of CSE, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Abhishek Sharma (191258)
Shivam Singh Negi (191260)

Abstract

As we all know that covid-19 has put a major setback on the Indian education system. The pandemic caused shocks to the system with schools forced to shut down during the lockdown period, and the transition of students and teachers to online teaching-learning. In India, around 250 million students were affected due to school closures at the onset of lockdown induced by COVID-19. There is a lack of seriousness among students while attending the online classes. Most of the students join the class just for attendance and thereafter go for their own works i.e playing games , sleeping. The teacher continues to teach without knowing what students are doing at the other end. It is just like talking to the walls.There is no point in teaching in such a way where no one is listening and in exams it's easy to just copy paste from the google. Also, if the teacher asks questions to each student one by one it would take a lot of time. Thus we can't really judge the performance of the students.

Online Student Activeness System is a model which can help the teachers to check the activeness of the students on a daily basis in the classroom. Teachers can evaluate the students in online mode just like they are physically present in the classroom. They can know how many students are actively present in the class and how many of them have joined the class for just the sake of attendance. It also helps the teachers to know how much students are able to understand from what he is teaching. Teacher will be getting the data of every student from time to time and from there he/she can easily sort out the students who are not active. It also stores the data onto a csv file on the teachers computer thereby for the future reference (for giving internal marks).

INDEX

==

List of figures

==

Chapter 04: Results figures:-

Chapter 01: INTRODUCTION

1.1 Introduction

As due to covid-19 almost all the education has been done online mode for the last 2 years. All the students attend the classes but most of them are not making effective use of online classes, they are just there for the sake of attendance . Most of the students join the class on their device , leave it as it is and get busy with their other activities. Teachers think that all of them are listening without knowing the actual catch behind the screen.Now , teachers can't ask each of the students to turn on their camera as it would take a lot of time. Such a system is a total waste of time and effort put by teachers.

Online Student Activeness System is a model created to help teachers get the daily basis performance of students in online classes. Here , the teacher need not to worry about whether the student is present behind the screen. The system will automatically tell the teacher the activeness of each student during the class without compromising their privacy.

After the teacher starts teaching, a few questions will pop up on the screen at any instant of time during the class hours. Teachers need not to worry about the question as questions will be asked randomly and any time to a student. Each student will be asked questions at random time so it's not like the student knows when the question will come and he/she comes to the screen and answers the question. At the end of class the teacher will get a record consisting of no of questions attempted corresponding to each student and that data will be directly stored in a csv file against the student Roll.No . The teacher will get the attributes as follows -:

- Accuracy- percentage of no of correct question out of questions attempted
- Missed- percentage of number of question missed by student
- Clicks - percentage of number of clicks by student at the time when question appears out of total no of clicks by student during class hour
- Scroll - percentage of scroll done by student at the time when question appears out of amount of scrolls by student during class hours.

The questions to be asked to students will be picked from a csv file that will be created by the teacher for each subject and will have multiple choice questions only. In this case a web scraper has been created to fetch questions from a website(www.sanfoundry.com) and have stored them in a csv file. Teacher can create a set of questions on his own or can also create a scraper to fetch questions.The CSV file created will be stored at the teacher system which will act as server and all the students will act as clients.

As considering the case that , no of attempted questions only will not just be sufficient to predict the activeness of students, we have added other parameters such as no of clicks and no of scrolls. We calculated the no of clicks by students during the class and the no of clicks when the question pops up. From that we calculated the percentage of clicks done by students when the pop up arrived. We calculated the same for the percentage of scrolls.

1.2 Objective

The main objective of the model are as follows:

- Check the activeness and performance of students in online classes. The system will ask the students a set of random questions selected from the database at any instant of time during the class. The total no of questions answered and no of questions answered correctly will be stored in an csv file against the student enrollment number.
- Calculate the effective clicks and effective scrolls done by student during the class and store them in a csv file.
- From the attributes (Accuracy %, Missed %, Clicks %, Scrolls %) , classify the students into four groups mainly: Highly Active (Level 1), Active (Level 2),Less Active(Level 3), Least Active(Level 4) using the machine learning algorithms such as KNN algorithm and Linear Support Vector Machine.

1.3 Motivation

As mentioned earlier , the impact of covid-19 on the education system has changed more than the way it was 3-4 years ago. Online Activity of students can't be directly monitored by the teacher. He / She can do it for at most 4-5 students in class and also in most of the cases the classes are for 1 hour and monitoring so many students can’t be done in such a short span of time. Thus during the class the teacher can’t examine the performance of the student. And it all comes to the exams which are also online and there is nothing which is not available on internet. Thus during exams students directly copy paste from internet and so can't determine the students performance based on such parameters. So, a method is in which a teacher can focus only on teaching during class and he/she could also get each student’s performance and that’s where our model comes into play.

1.4 Language Used

This model has been implemented using python .Various pre-built modules of python are used to complete this task

1.5 Technical Requirements (Hardware)

- Mouse and keyboard are essential for input and data collection
- Any python supporting IDE should be installed in the system.
- Since multiprocessing is used there is need to either have a system having multiple processors or the processor should have multiple cores.

1.6 Deliverables/Outcomes

The outcome of the project is a monitoring ,predicting and assessment system that monitors students on a daily basis and classifies them into four groups : Highly Active (Level 1), Active (Level 2),Less Active(Level 3), Least Active(Level 4) . All the data will be stored in an CSV file so the teachers can access the records in future and assess the students(i.e used the figures to give marks to the students

Chapter 02: Feasibility Study, Requirements Analysis and Design

==

2.1 Feasibility Study

Since a model is created , there are various assumptions that have been made in order to accomplish the task. Below are the assumptions made by :

- The duration of the class is 1 hour(it could be extended as per teachers requirement)
- Only Multiple choice questions are asked to the students
- The number of clicks are proportional to the number of input from the keyboard so it has been ignored.
- There is no time delay between the system local time and application time(i.e. both are in sync with each other).
- Internet connectivity is good
- There is no conflict in accessing the file by various clients.

2.1.1 Problem definition and Analysis

a)Input from keyboard:The assumption is whenever there is a click there is also an entry from the keyboard and hence both will have a similar impact on the dataset. But in reality it could exist that the no number of clicks and entries from the keyboard are not the same. It could be possible that the student is using the keyboard keys such as arrow and enter rather than using a mouse. In this case these entries would not be taken into account

Solution : We can use the **pynput.keyboard()** module of the python to calculate total no of keys pressed during the class and and no of keys pressed during the pop up. Then we can calculate the percentage of keys pressed during the pop up and add as an attribute for classifying in the CSV file.

b)Technical Feasibility: It could be possible that the system used by the student does not support multiprocessing . Then the system will go out of sync as the resource will be occupied by one process until it doesn't finishes. Then the pop up will not be able to sync with the class.

Solution: Student use a system which supports multiprocessing

2.2 Requirements

2.2.1 Functional Requirements

- The system automatically pops up the questions at any instant of time during the class hours to the student and the result is stored in a file.
- System automatically calculates the no of clicks and scrolls from the student
- The system automatically classifies the students into groups based on the values of each attribute stored in the csv file .
- Data received from various students is being stored in an CSV file and is not being overwritten.
- Python modules required for proper function of the application are available

2.1.2 Non-Functional Requirements

- System on which application is running supports multiprocessing
- The questions to be asked are directly fetched from the teacher's end . The dataset of the question is available online on the teachers system
- The system is sending the data i.e report of the students to the teacher after the class ends or the application is terminated.

Chapter 03: IMPLEMENTATION

===

3.1 Modules Used

Various modules of python to calculate attributes of the CSV file. The Pycharm IDE has been used to implement the project and in few cases we have also used Google Colab for implementing python scripts.The modules used are:

- **pandas** : It is an open-source library that is made mainly for working with relational or labelled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series.
- **tkinter** : It offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create GUI applications. We have used it to create a pop up for displaying questions.
- **random**: It's a built-in module that you can use to make random numbers. It is used to get random time for displaying pop up.
- **multiprocessing** : It includes a very simple and intuitive API for dividing work between multiple processes. It is used to run pop up and pynput module simultaneously.
- **datetime** : supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.
- **pynput** : This library allows you to control and monitor input devices. We have used it to calculate the number of clicks and scrolls.
- **loggings:** It is a means of tracking events that happen when some software runs.

3.2 Algorithm for Project (Theory)

First of all when the teacher starts the class and a student enters the class. Each student has a unique id which is Roll. No against which his/her record will be stored . Since each student will

have a different roll no thus it can act as an primary key and there will be no clash among storing records of each student.when the teacher starts teaching the student will be in front of the screen , a pop up will be displayed on the student screen at any instant of time in class.The pop up has been created with the help of **tkinter** module of python which will have a question randomly picked up from the question database. In our case this pop up will be displayed 6 times as we have considered that class is of 1 hour and each question will be asked after an interval of time. For example: If class starts at 9:00 Am , then the first pop up will arrive between 9:00-9:10 Am , second will be displayed between 9:11-9:20 Am , then accordingly the pop ups will be displayed at an interval of 10 mins.
If the student is on the screen then certainly he will see the pop up and will try to answer the question and if he is not on the screen then he/she will miss the question. Each student will be given a time of 2 min to answer the question. In an interval of time if the student knows the answer then he will instantly answer the question or else he will google and will try to attempt the question. Now after the time interval is over if the answer of the question is correct the the no of attempted_questions will be incremented by 1 and correct_question will be incremented by 1 and if the answer is wrong then then wrong_questions will be incremented by 1.If the student fails to answer the question then the missed_factor will be incremented by 1. This step will be repeated for total no of question time which is 6 in our case. After this we will get the final values of attempted_question, correct_question, wrong_question, missed_facor. Then we will calculate the Accuracy_percentage, Missed_percentage and will store the values in the CSV file against the Roll.No of students.

Accuracy_percentage, Missed_percentage will be calculated as follows:

- Accuracy_percentage=(correct_questions/attempted_question)*100
- Missed_percentage=(missed_question/6)*100

Now it could be possible that the student has opened multiple tabs and on one he/she is attending class and on he/she is doing some other game like playing games, watching movies etc. Then he/she can easily get to know when the question arrives and will answer it. Considering this case we have used the **pynput** module of python and have calculated the total no of clicks and no of scrolls during the class. As mentioned , the pynput module stores the clicks and scroll at a time in a text file.We calculate the number of Clicks_number and Scroll_number from the text file. Then from the text file we calculate the no clicks and scroll at the time when the pop up is displayed. Then we Clicks_percentage and Scroll_percentage and store them in a CSV file.

Clicks_percentage and Scroll_percentage are calculated as follows:

- Scroll_percentage=((no of scroll at pop up)/Clicks_number)*100
- Clicks_percentage=((no of clicks at pop up)/Clicks_number)*100

The pop up function and the pynput function will run at the same time. Since these two are different processes and in order to implement them simultaneously we need to use the concept of multiprocessing. In order to use multithreading we have used the **multiprocessing** module of python.

After the application ends the client system sends the all above mentioned attributes to the server i.e. teacher system in the format of string . After the server receives the string it splits the string using split(",") function and "," act as a separator.

The server stores all the values in the CSV file against the ROll.No of the student. Now at the server end it need to classify the student into four groups:

- Highly Active (Level 1)
- Active (Level 2)
- Less Active(Level 3)
- Least Active(Level 4)

It uses a machine learning classification algorithm to classify the student into groups. We have used two machine learning classification algorithms:

- K-Nearest Neighbour Algorithm
- Support Vector Machine Algorithm

In order to classify them first of all we have assigned weight to each attribute. Accuracy_percentage,Missed_percentage, Clicks_percentage, Scroll_percentage are assigned weights of 10, 30 , 30 , 30 respectively. For each student we have calculated the w_Accuracy, w_Missed, w_Clicks , w_Scroll and store them in file.

w_Accuracy, w_Missed, w_Clicks , w_Scroll are calculated as follow:

- w_Accuracy=Accuracy_percentage*0.1
- w_Missed=(100-(Missed_percentege*0.3)
- w_Clicks=Clicks_percentage*0.3
- w_Scroll=Scroll_percentage*0.3

Then we calculate the sum of w_Accuracy, w_Missed, w_Clicks , w_Scroll for each student and store it in variable Sum.

Now , we have classified the students on the basis of the sum.

If the sum is greater than 75 then the student is classified as Highly Active (Level 1)

If sum is in between 50 and 74 , then student is classified as Active (Level 2)

If sum is in between 25 and 49 , then student is classified as Less Active (Level 3)

If sum is less than 25 , then student is classified as Least Active(Level 4)

Then the Result is stored as an attribute in the CSV file.

3.3 E-R Diagram / Data-Flow Diagram (DFD)

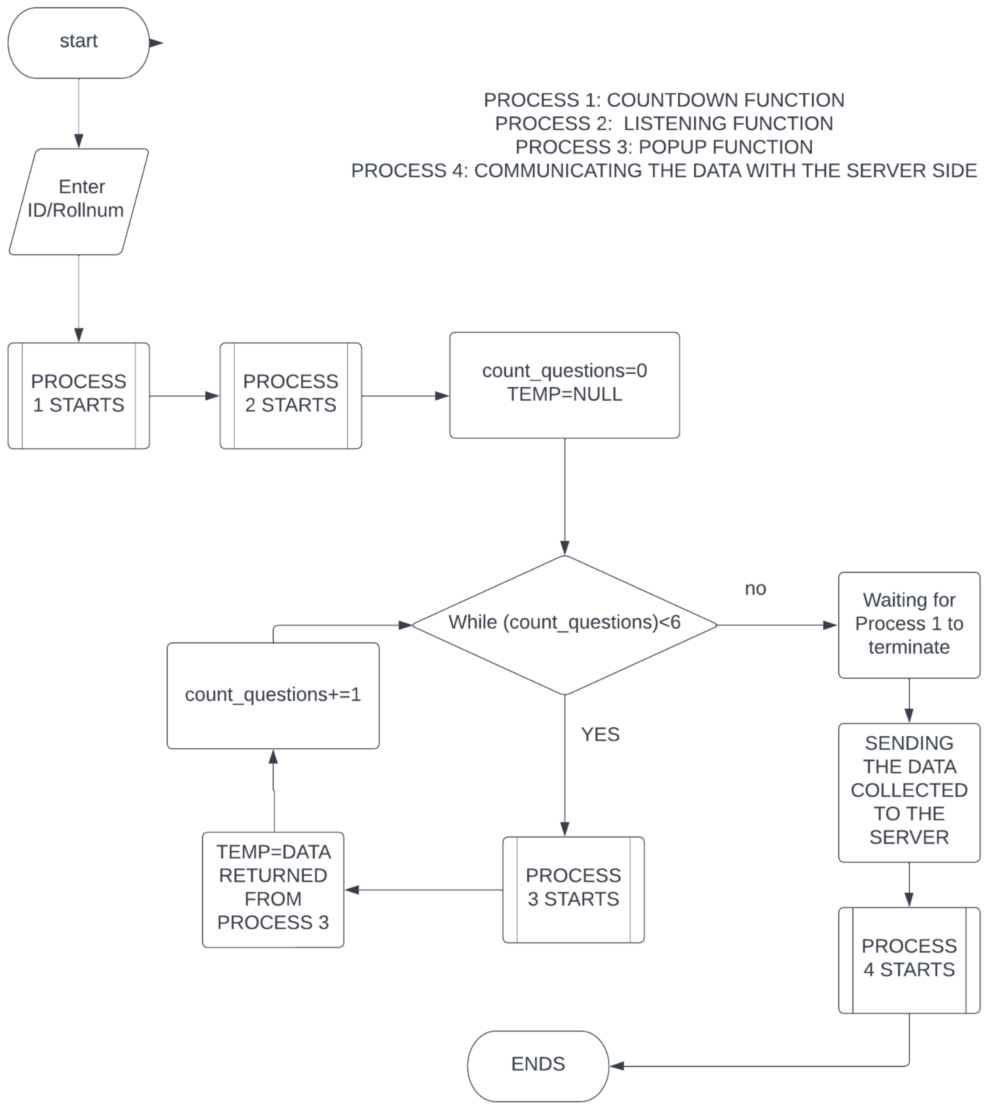

start
PROCESS 1: COUNTDOWN FUNCTION
PROCESS 2: LISTENING FUNCTION
PROCESS 3: POPUP FUNCTION
PROCESS 4: COMMUNICATING THE DATA WITH THE SERVER SIDE
Enter ID/Rollnum
PROCESS 1 STARTS
PROCESS 2 STARTS
count_questions=0 TEMP=NULL
While (count_questions)<6
no
Waiting for Process 1 to terminate
count_questions+=1
YES
SENDING THE DATA COLLECTED TO THE SERVER
TEMP=DATA RETURNED FROM PROCESS 3
PROCESS 3 STARTS
PROCESS 4 STARTS
ENDS

Fig 3.1:Main flow at the user end

Fig 3.1 shows the main flow of the program at the user/student end. First of all, the system asks the user to enter his/her Roll.No which is unique for each user. Then the system starts the Process 1 which is a countdown function which counts the time (similar to clock) and tells when the class is over. Then Process 2 is started which records the no of clicks and scroll and stores it in a text file.Then count_questions is a variable which tells no of questions displayed (at max 6 in this case) and is initially initialised to zero. Then the if else statement is executed until all questions are displayed. If count_questions is less than 6 then process 3 which pops up is displayed and the result is stored in the temp variable and count_questions is incremented by 1. When the loop ends, the system waits for class to finish, i.e process 1 to end. After the class terminates, the recorded data is sent to the server for storing in a CSV file.

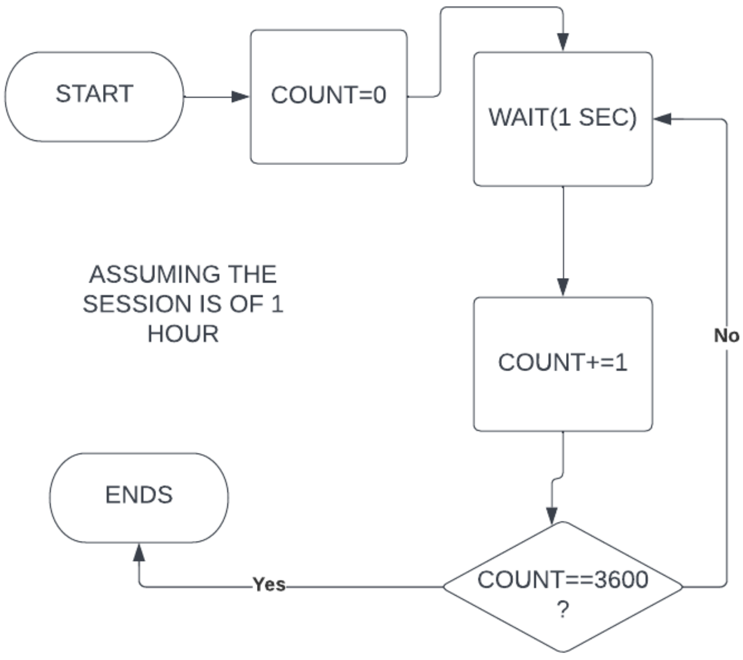


Fig 3.2.Flowchart of PROCESS 1

Fig 3.2 is a function for calculating the time . It is considered that class is of 1 hour and number of seconds are calculated as count variables. Initially count is initialised to 0. Then we wait for 1 sec and increment count by 1. Later the loop runs until the count is less than 3600(1 hour=60 mins= 3600 sec) . When time is over ,the process is terminated

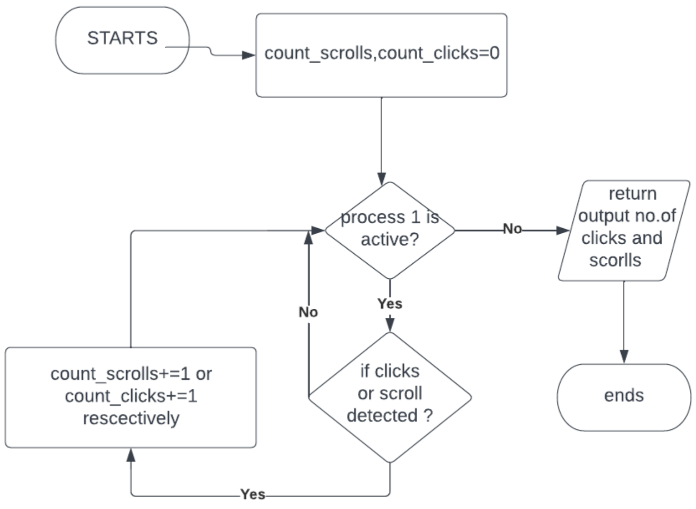

STARTS
count_scrolls,count_clicks=0
process 1 is active?
No
return output no.of clicks and scorlls
ends
Yes
No
if clicks or scroll detected ?
count_scrolls+=1 or count_clicks+=1 resceсtively
Yes

Fig3.3: Flowchart of PROCESS 2

Fig 3.3 is for calculating the number of clicks and scroll by the user. Initially count_clicks and count_scrolls are initialised to 0. Until the process 1 is not terminated , if scroll or click is detected then count_clicks and count_scrolls are incremented by 1 respectively. This is done using pynput which is pre pre-built library in python. When the class terminates count_clicks and count_scrolls are returned.

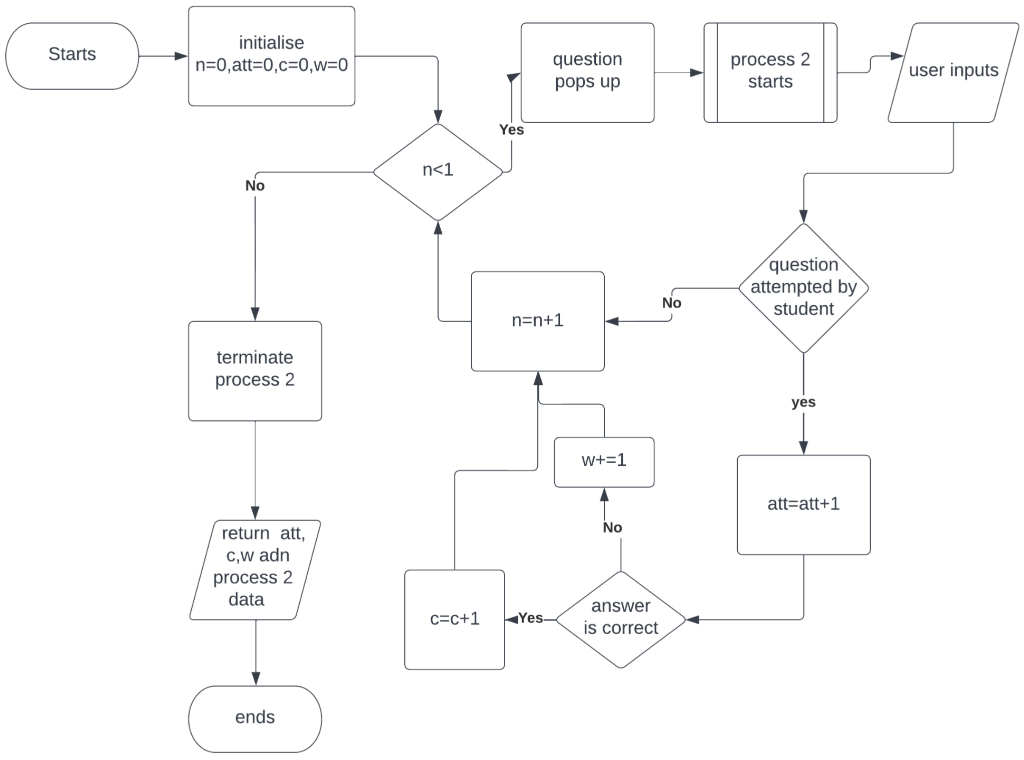


Fig 3.4: Flowchart of PROCESS 3

Fig 3.4 is a pop up function which picks questions from a dataset and displays them as popup. Here the variables shows following details:

- n= number of questions
- att= number of attempted questions
- c=number of correct questions
- w=number of wrong questions

Initially all of them are stored to 0. Since this function is called from the main function, this pop up will be called once. While n is less than 1 , pop up will be displayed , now there could be two possible cases:

- If a student attempts the question, then att will be incremented by 1 and the answer is correct then c is incremented by 1 and if answer is wrong then w is incremented by 1. Then n is incremented by 1 and comes out of 1.
- If a student misses a question then simply n is incremented.

At last, when the values of the variables are returned.

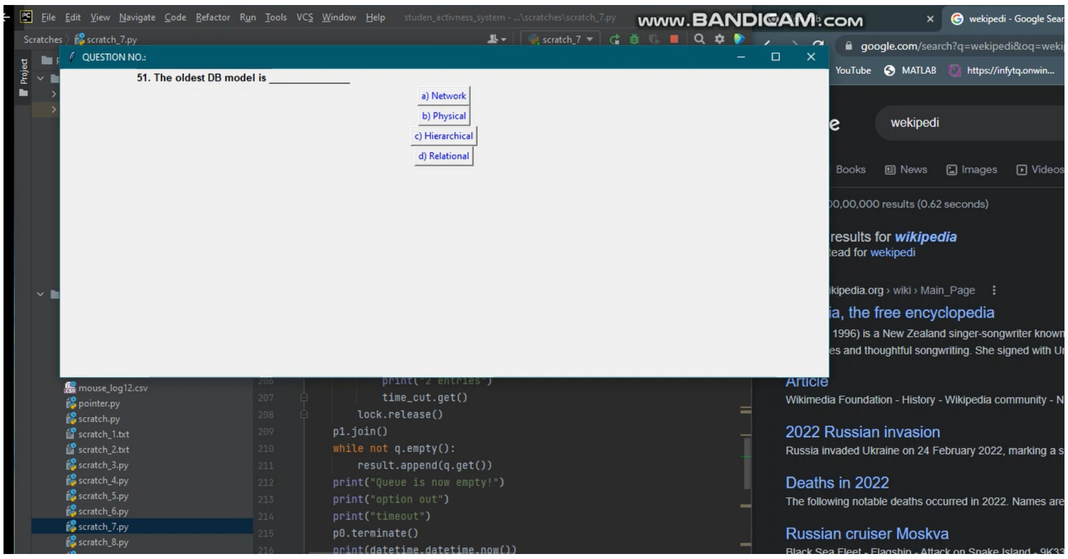
At the Server side

Server receives the data from various clients.

- Then all the data will be collected and stored in a single common file (mostly the csv).
- Then we will analyse the data and then classify the students into four groups using the machine learning algorithms such as KNN and Linear Support Vector Machine and finally find the accuracy.
-

3.3 Algorithm for Project (Practical Implementation)

3.3.1- popup() with outputs



QUESTION NO.:

51. The oldest DB model is ____________

a) Network

b) Physical

c) Hierarchical

d) Relational

```
            print("2 entries")
            time_cut.get()
        lock.release()
    p1.join()
    while not q.empty():
        result.append(q.get())
    print("Queue is now empty!")
    print("option out")
    print("timeout")
    p0.terminate()
    print(datetime.datetime.now())
```

00,00,000 results (0.62 seconds)

results for *wikipedia*
ead for wekipedi

ikipedia.org › wiki › Main_Page

ia, the free encyclopedia
1996) is a New Zealand singer-songwriter known
es and thoughtful songwriting. She signed with U

Article
Wikimedia Foundation - History - Wikipedia community - N

2022 Russian invasion
Russia invaded Ukraine on 24 February 2022, marking a s

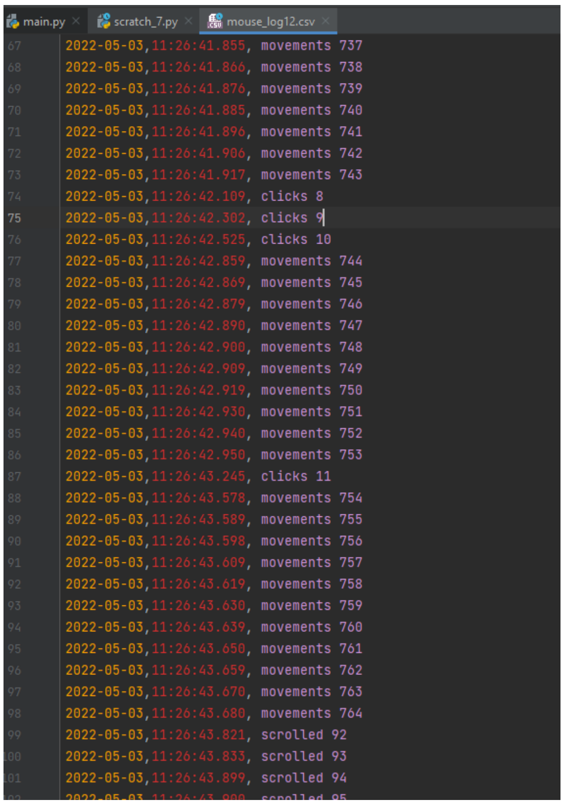
Deaths in 2022
The following notable deaths occurred in 2022. Names are

Russian cruiser Moskva
Black Sea Fleet - Flagship - Attack on Snake Island - 9K33

Fig 3.5:Pop being displayed to student

Fig 3.5 shows the pop up being displayed to the user at any random time during the class . This window will be available for 2 minutes for the user to answer

3.3.2-clicks and scroll recording function-

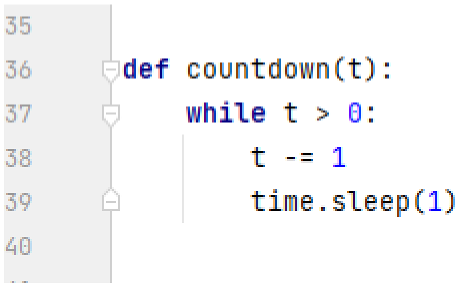


```
2022-05-03,11:26:41.855, movements 737
2022-05-03,11:26:41.866, movements 738
2022-05-03,11:26:41.876, movements 739
2022-05-03,11:26:41.885, movements 740
2022-05-03,11:26:41.896, movements 741
2022-05-03,11:26:41.906, movements 742
2022-05-03,11:26:41.917, movements 743
2022-05-03,11:26:42.109, clicks 8
2022-05-03,11:26:42.302, clicks 9
2022-05-03,11:26:42.525, clicks 10
2022-05-03,11:26:42.859, movements 744
2022-05-03,11:26:42.869, movements 745
2022-05-03,11:26:42.879, movements 746
2022-05-03,11:26:42.890, movements 747
2022-05-03,11:26:42.900, movements 748
2022-05-03,11:26:42.909, movements 749
2022-05-03,11:26:42.919, movements 750
2022-05-03,11:26:42.930, movements 751
2022-05-03,11:26:42.940, movements 752
2022-05-03,11:26:42.950, movements 753
2022-05-03,11:26:43.245, clicks 11
2022-05-03,11:26:43.578, movements 754
2022-05-03,11:26:43.589, movements 755
2022-05-03,11:26:43.598, movements 756
2022-05-03,11:26:43.609, movements 757
2022-05-03,11:26:43.619, movements 758
2022-05-03,11:26:43.630, movements 759
2022-05-03,11:26:43.639, movements 760
2022-05-03,11:26:43.650, movements 761
2022-05-03,11:26:43.659, movements 762
2022-05-03,11:26:43.670, movements 763
2022-05-03,11:26:43.680, movements 764
2022-05-03,11:26:43.821, scrolled 92
2022-05-03,11:26:43.833, scrolled 93
2022-05-03,11:26:43.899, scrolled 94
2022-05-03,11:26:43.900, scrolled 95
```

Fig 3.6: File in which output of pynput module is stored

Fig 3.6 shows the file that contains the output from listener function i.e listener function , which stores the click, scroll or movement in a text file.

3.3.3 Timer/countdown Function

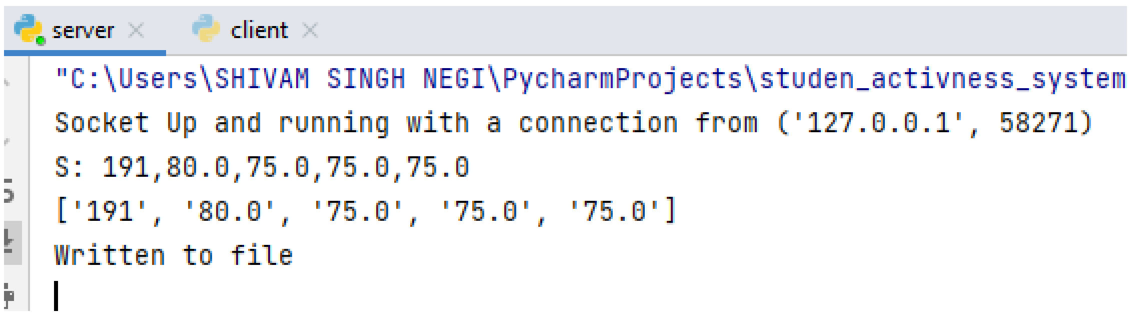


```python
def countdown(t):
    while t > 0:
        t -= 1
        time.sleep(1)
```

Fig 3.7: Timer function

Fig 3.7 shows a countdown timer for class

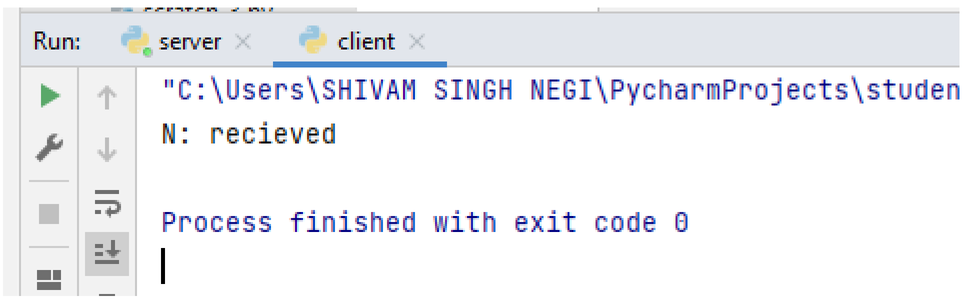
3.3.4 client server model implementation:



```
"C:\Users\SHIVAM SINGH NEGI\PycharmProjects\studen_activness_system
Socket Up and running with a connection from ('127.0.0.1', 58271)
S: 191,80.0,75.0,75.0,75.0
['191', '80.0', '75.0', '75.0', '75.0']
Written to file
```

Fig 3.8:Server's end

Figure 3.8 shows data collected at the client end being sent to the server in the form of string . The server accepts the sting and splits it using " , " as a separator and store it in a csv file



```
"C:\Users\SHIVAM SINGH NEGI\PycharmProjects\studen
N: recieved

Process finished with exit code 0
```

Fig 3.9:Client's end

Fig 3.9 shows that when the receiver receives the data from the client it sends an acknowledgement back in form of a string " received" .

3.4 Data Set Used

We have created our own dataset for this project. As we could not find anything related to this on the internet so we had to create our own data set .

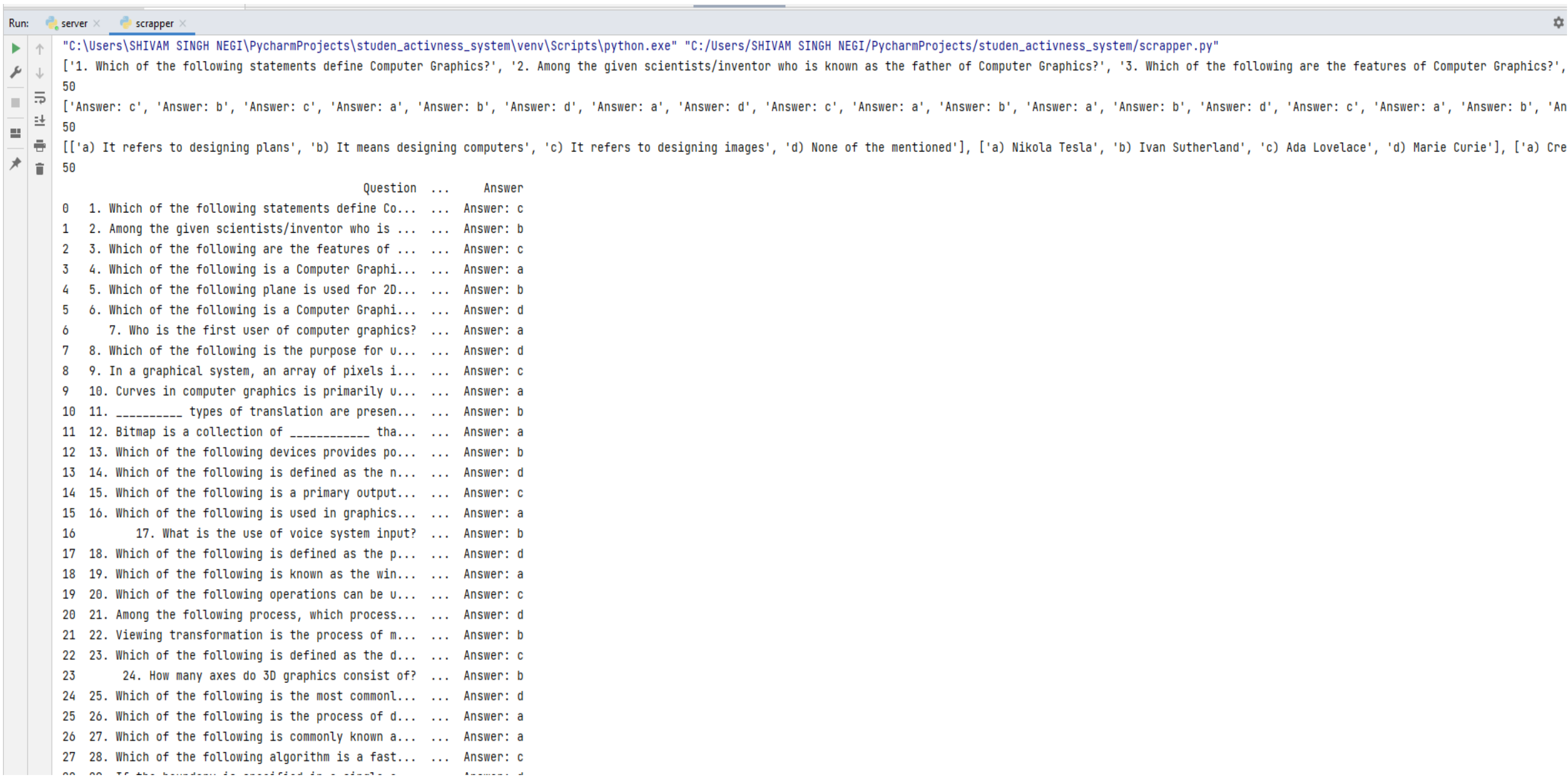
We have created two types of datasets:

1. By using a web scraper we have created a dataset of questions for a particular topic which can be used to ask the questions to students in pop up.
2. The dataset has been created for implementing this project which consists of all possible test cases . In that dataset we have stored the values corresponding to each student and furthermore we have classified them into groups

3.4.1 Web Scraper Dataset

It consists of set of question extracted from a website(www.sanfoundry.com) with the help of scraper to ask questions from students

Output after extracting the questions from website



```
"C:\Users\SHIVAM SINGH NEGI\PycharmProjects\studen_activness_system\venv\Scripts\python.exe" "C:/Users/SHIVAM SINGH NEGI/PycharmProjects/studen_activness_system/scrapper.py"
['1. Which of the following statements define Computer Graphics?', '2. Among the given scientists/inventor who is known as the father of Computer Graphics?', '3. Which of the following are the features of Computer Graphics?',
50
['Answer: c', 'Answer: b', 'Answer: c', 'Answer: a', 'Answer: b', 'Answer: d', 'Answer: a', 'Answer: d', 'Answer: c', 'Answer: a', 'Answer: b', 'Answer: a', 'Answer: b', 'Answer: d', 'Answer: c', 'Answer: a', 'Answer: b', 'An
50
[['a) It refers to designing plans', 'b) It means designing computers', 'c) It refers to designing images', 'd) None of the mentioned'], ['a) Nikola Tesla', 'b) Ivan Sutherland', 'c) Ada Lovelace', 'd) Marie Curie'], ['a) Cre
50
                                           Question  ...     Answer
0   1. Which of the following statements define Co...  ...  Answer: c
1   2. Among the given scientists/inventor who is ...  ...  Answer: b
2   3. Which of the following are the features of ...  ...  Answer: c
3   4. Which of the following is a Computer Graphi...  ...  Answer: a
4   5. Which of the following plane is used for 2D...  ...  Answer: b
5   6. Which of the following is a Computer Graphi...  ...  Answer: d
6      7. Who is the first user of computer graphics?  ...  Answer: a
7   8. Which of the following is the purpose for u...  ...  Answer: d
8   9. In a graphical system, an array of pixels i...  ...  Answer: c
9   10. Curves in computer graphics is primarily u...  ...  Answer: a
10  11. __________ types of translation are presen...  ...  Answer: b
11  12. Bitmap is a collection of ____________ tha...  ...  Answer: a
12  13. Which of the following devices provides po...  ...  Answer: b
13  14. Which of the following is defined as the n...  ...  Answer: d
14  15. Which of the following is a primary output...  ...  Answer: c
15  16. Which of the following is used in graphics...  ...  Answer: a
16         17. What is the use of voice system input?  ...  Answer: b
17  18. Which of the following is defined as the p...  ...  Answer: d
18  19. Which of the following is known as the win...  ...  Answer: a
19  20. Which of the following operations can be u...  ...  Answer: c
20  21. Among the following process, which process...  ...  Answer: d
21  22. Viewing transformation is the process of m...  ...  Answer: b
22  23. Which of the following is defined as the d...  ...  Answer: c
23       24. How many axes do 3D graphics consist of?  ...  Answer: b
24  25. Which of the following is the most commonl...  ...  Answer: d
25  26. Which of the following is the process of d...  ...  Answer: a
26  27. Which of the following is commonly known a...  ...  Answer: a
27  28. Which of the following algorithm is a fast...  ...  Answer: c
```

Fig 3.10:File in which questions are stored

Fig 3.10 shows the question being fetched from a external site www.sanfoundry.com with the help of web scraper and printing the questions fetched.

3.4.2 Students Dataset

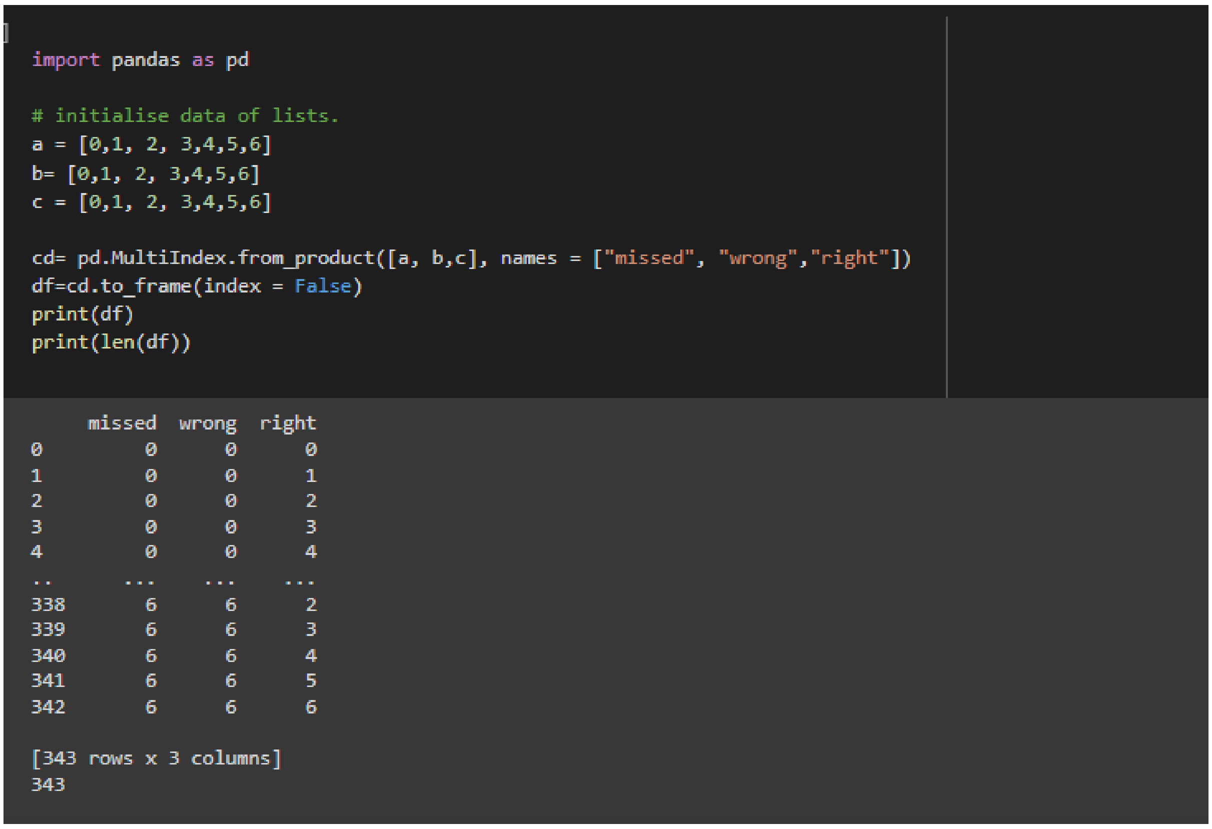
This dataset consists of all possible test cases for our project. We have considered that 6 questions will be asked from students and according to that we have calculated all the possible test cases and have classified them into groups by the algorithm as mentioned above.

The attributes of the dataset are:

- Roll.No(Roll No of student)
- Accuracy_percentage(Percentage of correct answers out of attempted questions)
- Missed_factor (Percentage of missed questions)

- Clicks_percentage(Percentage of Clicks during pop up)
- Scroll_percentage(Percentage of Scroll during pop up)
- w_Accuracy(Weighted Effective Accuracy)
- w_Missed(Weighted Effective Missed factor)
- w_Clicks(Weighted Effective Clicks)
- w_Scroll(Weighted Effective Scrolls)
- Result (Type of student: classified by machine learning algorithms)

3.4.2.1 Creation of dataset

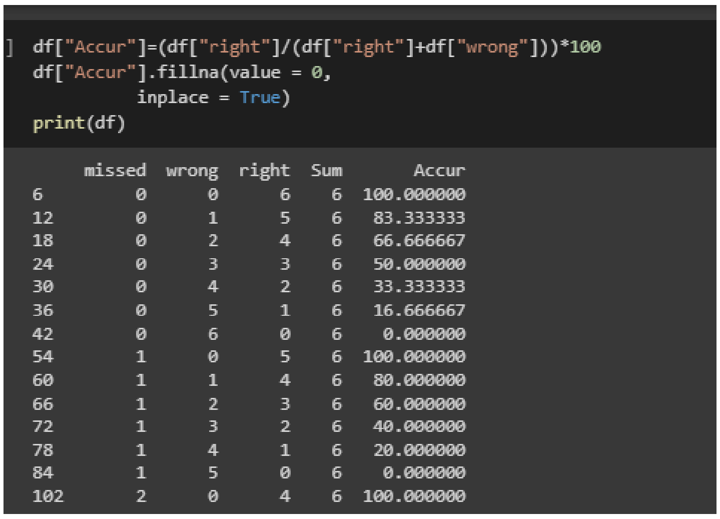


```python
import pandas as pd

# initialise data of lists.
a = [0,1, 2, 3,4,5,6]
b= [0,1, 2, 3,4,5,6]
c = [0,1, 2, 3,4,5,6]

cd= pd.MultiIndex.from_product([a, b,c], names = ["missed", "wrong","right"])
df=cd.to_frame(index = False)
print(df)
print(len(df))
```

```
     missed  wrong  right
0         0      0      0
1         0      0      1
2         0      0      2
3         0      0      3
4         0      0      4
..      ...    ...    ...
338       6      6      2
339       6      6      3
340       6      6      4
341       6      6      5
342       6      6      6

[343 rows x 3 columns]
343
```

Fig 3.11: Creating all possible combination

Fig 3.11 shows that all possible combinations are made for attempted and missed questions that if 6 questions are attempted all possibles outcomes that can be extracted are shown

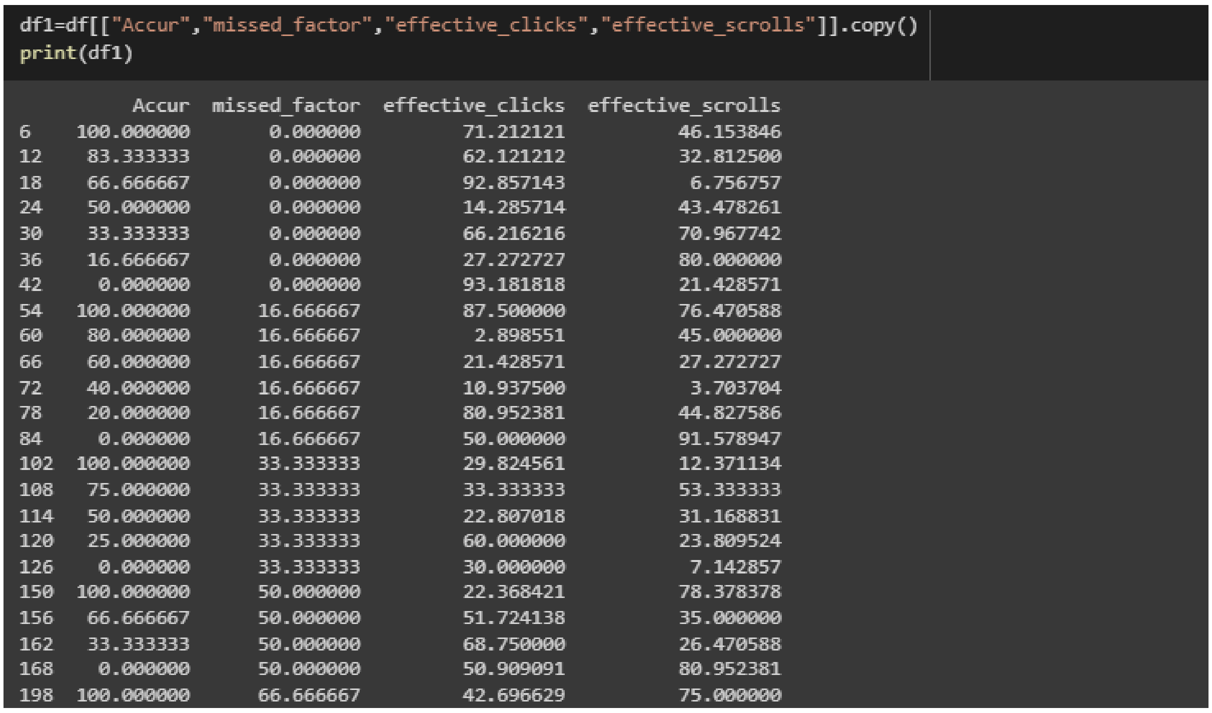


```
] df["Accur"]=(df["right"]/(df["right"]+df["wrong"]))*100
  df["Accur"].fillna(value = 0,
            inplace = True)
  print(df)
```

```
     missed  wrong  right  Sum       Accur
6         0      0      6    6  100.000000
12        0      1      5    6   83.333333
18        0      2      4    6   66.666667
24        0      3      3    6   50.000000
30        0      4      2    6   33.333333
36        0      5      1    6   16.666667
42        0      6      0    6    0.000000
54        1      0      5    6  100.000000
60        1      1      4    6   80.000000
66        1      2      3    6   60.000000
72        1      3      2    6   40.000000
78        1      4      1    6   20.000000
84        1      5      0    6    0.000000
102       2      0      4    6  100.000000
```

Fig 3.12: Calculating the Accuracy Percentage

Fig 3.12 shows calculating the accuracy of attempted questions by dividing the right by attended questions.

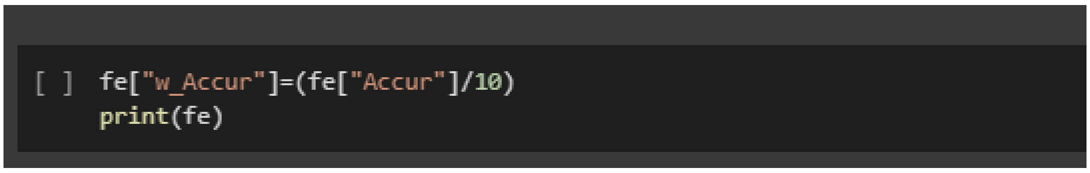


```
df1=df[["Accur","missed_factor","effective_clicks","effective_scrolls"]].copy()
print(df1)
```

```
          Accur  missed_factor  effective_clicks  effective_scrolls
6    100.000000       0.000000         71.212121          46.153846
12    83.333333       0.000000         62.121212          32.812500
18    66.666667       0.000000         92.857143           6.756757
24    50.000000       0.000000         14.285714          43.478261
30    33.333333       0.000000         66.216216          70.967742
36    16.666667       0.000000         27.272727          80.000000
42     0.000000       0.000000         93.181818          21.428571
54   100.000000      16.666667         87.500000          76.470588
60    80.000000      16.666667          2.898551          45.000000
66    60.000000      16.666667         21.428571          27.272727
72    40.000000      16.666667         10.937500           3.703704
78    20.000000      16.666667         80.952381          44.827586
84     0.000000      16.666667         50.000000          91.578947
102  100.000000      33.333333         29.824561          12.371134
108   75.000000      33.333333         33.333333          53.333333
114   50.000000      33.333333         22.807018          31.168831
120   25.000000      33.333333         60.000000          23.809524
126    0.000000      33.333333         30.000000           7.142857
150  100.000000      50.000000         22.368421          78.378378
156   66.666667      50.000000         51.724138          35.000000
162   33.333333      50.000000         68.750000          26.470588
168    0.000000      50.000000         50.909091          80.952381
198  100.000000      66.666667         42.696629          75.000000
```

Fig 3.13: Printing intermediate dataset

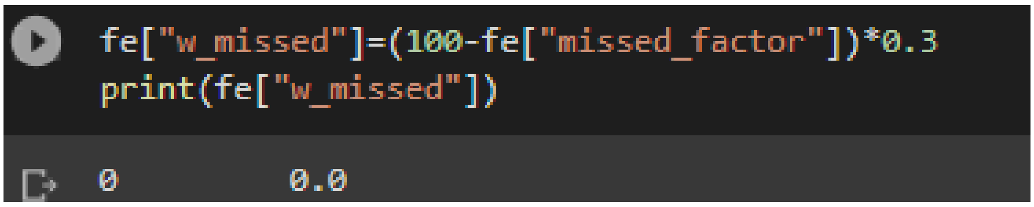
Fig 3.13 show all the attributes of the csv file in form of percentage



```
[ ] fe["w_Accur"]=(fe["Accur"]/10)
    print(fe)
```

Fig 3.14: Calculating weighted Accuracy

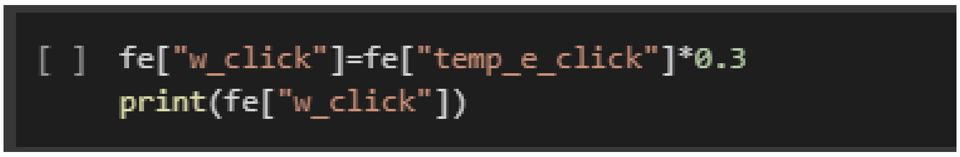
In Fig 3.14 weight of 10 is being assigned to the column Accuracy so that it can act as an parameter for classifying



```
fe["w_missed"]=(100-fe["missed_factor"])*0.3
print(fe["w_missed"])

0          0.0
```

Fig 3.15: Calculating weighted missed factor

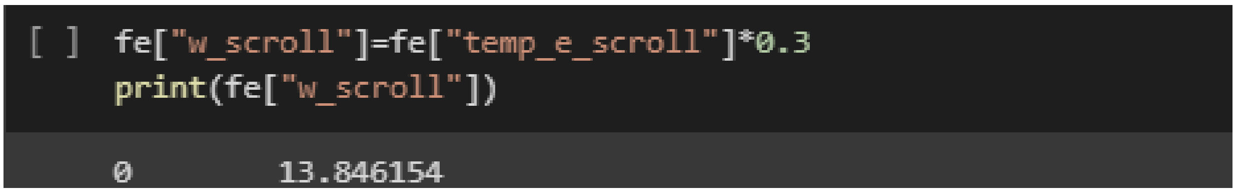
In Fig 3.15 weight of 30 is being assigned to the column missed so that it can act as a parameter for classifying.



```
[ ] fe["w_click"]=fe["temp_e_click"]*0.3
    print(fe["w_click"])
```

Fig 3.16: Calculating weighted clicks

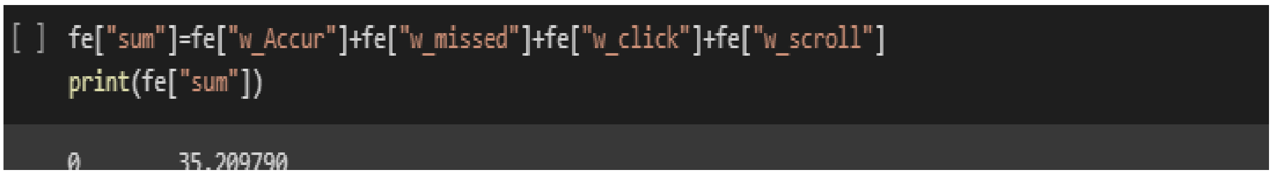
In Fig 3.16 weight of 30 is being assigned to the column clicks so that it can act as a parameter for classifying.



```
[ ] fe["w_scroll"]=fe["temp_e_scroll"]*0.3
    print(fe["w_scroll"])

    0          13.846154
```

Fig 3.17: Calculating weighted Scrolls

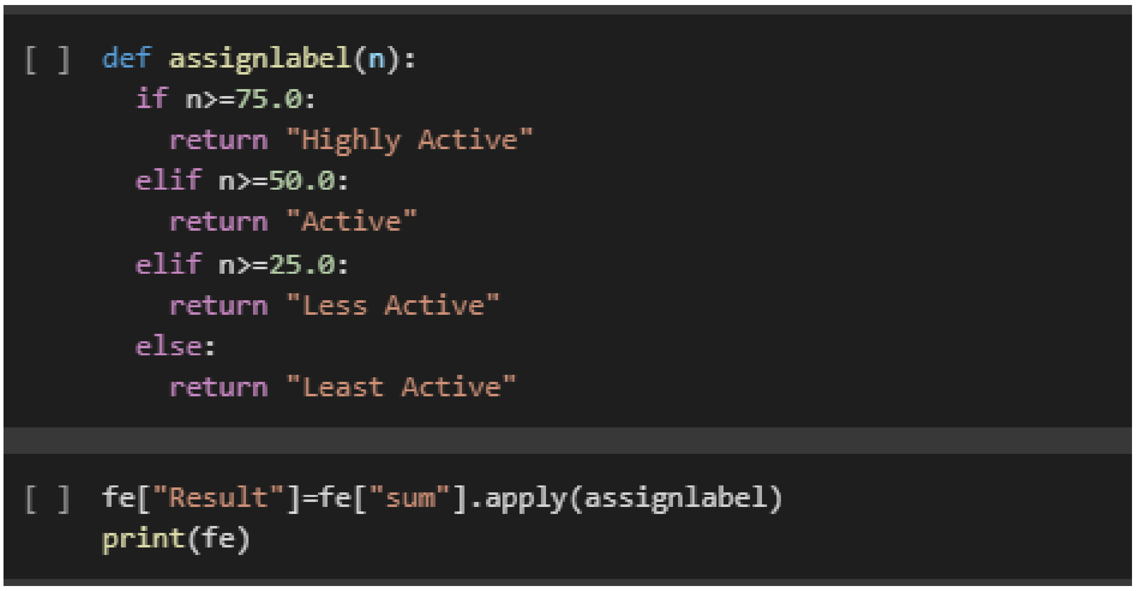
In Fig 3.17 weight of 30 is being assigned to the column scroll so that it can act as a parameter for classifying.



```
[ ] fe["sum"]=fe["w_Accur"]+fe["w_missed"]+fe["w_click"]+fe["w_scroll"]
    print(fe["sum"])

    0        35.209790
```

Fig 3.18: Calculating attribute Sum

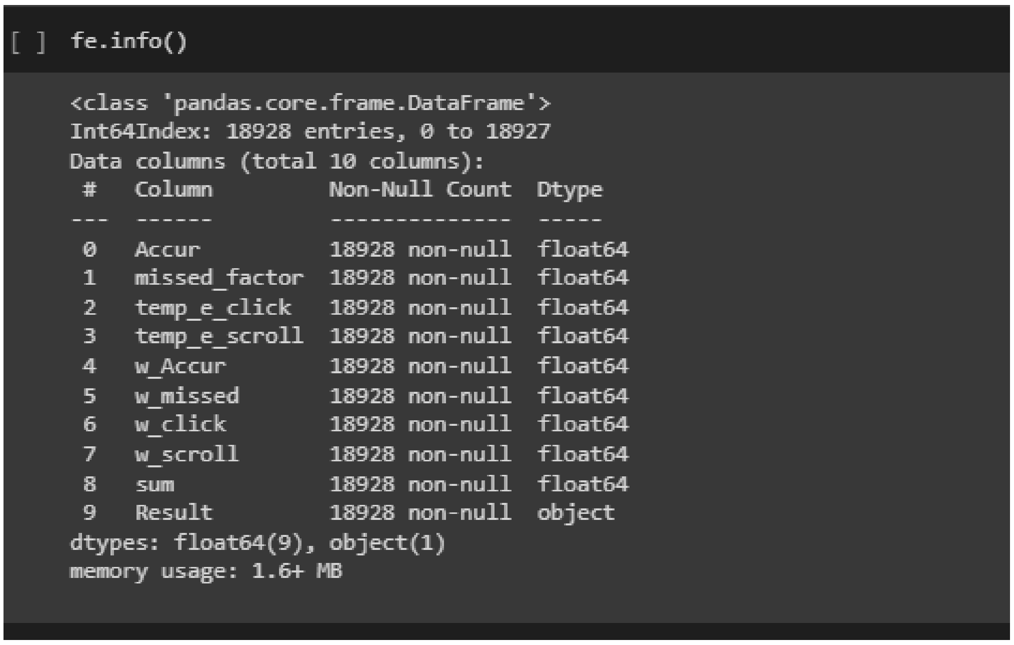
In Fig 3.18, the sum of all the parameters is taken and is stored as an attribute in the dataframe.



```
[ ] def assignlabel(n):
      if n>=75.0:
        return "Highly Active"
      elif n>=50.0:
        return "Active"
      elif n>=25.0:
        return "Less Active"
      else:
        return "Least Active"

[ ] fe["Result"]=fe["sum"].apply(assignlabel)
    print(fe)
```

Fig 3.19: Classifying students into groups

In Fig 3.19 , the users are classified into groups based on their sum.



```
[ ] fe.info()

    <class 'pandas.core.frame.DataFrame'>
    Int64Index: 18928 entries, 0 to 18927
    Data columns (total 10 columns):
     #   Column         Non-Null Count  Dtype
    ---  ------         --------------  -----
     0   Accur          18928 non-null  float64
     1   missed_factor  18928 non-null  float64
     2   temp_e_click   18928 non-null  float64
     3   temp_e_scroll  18928 non-null  float64
     4   w_Accur        18928 non-null  float64
     5   w_missed       18928 non-null  float64
     6   w_click        18928 non-null  float64
     7   w_scroll       18928 non-null  float64
     8   sum            18928 non-null  float64
     9   Result         18928 non-null  object
    dtypes: float64(9), object(1)
    memory usage: 1.6+ MB
```

Fig 3.20: Description of final dataset

In Fig 3.20 all the final attributes of the final dataset and their type are shown .

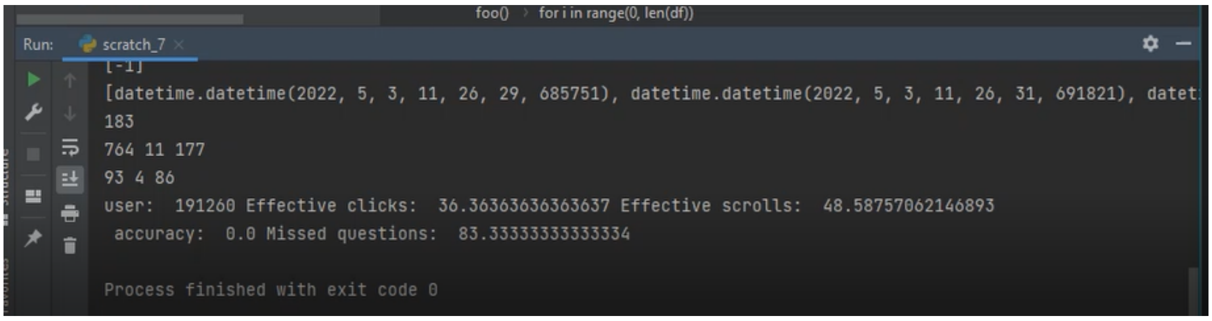
Chapter 04: RESULTS

===

4..1 Results

We are able to determine the activeness of the student during the online session/classes based on the result of the MCQ and the number of Clicks and Scroll During the pop up.

4.1.1 After the class is over and all pop ups are displayed

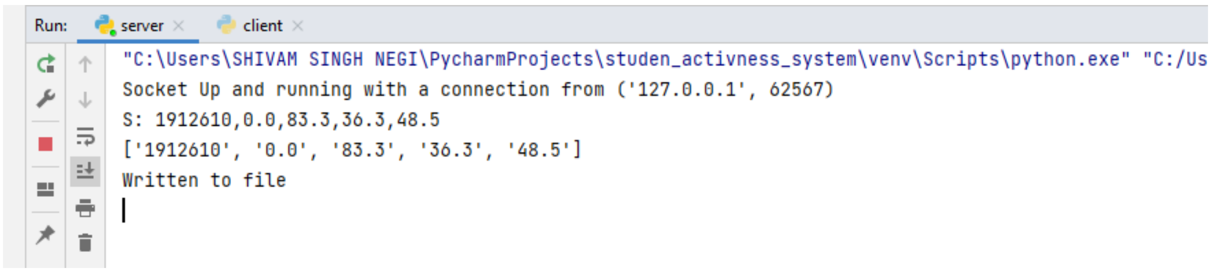


```
[-1]
[datetime.datetime(2022, 5, 3, 11, 26, 29, 685751), datetime.datetime(2022, 5, 3, 11, 26, 31, 691821), datet
183
764 11 177
93 4 86
user:  191260 Effective clicks:  36.36363636363637 Effective scrolls:  48.58757062146893
 accuracy:  0.0 Missed questions:  83.33333333333334

Process finished with exit code 0
```

Fig4.1: Result of student for popup

In Fig 4.1 3 questions are asked to the user and he/she has a time interval of 10s to answer . Out of the 3 questions, students attempt 1 question and it is answered incorrectly . So , -1 is appended to the tuple , for correct answer 1 is appended.
This result will go to server and will be stored in a csv file

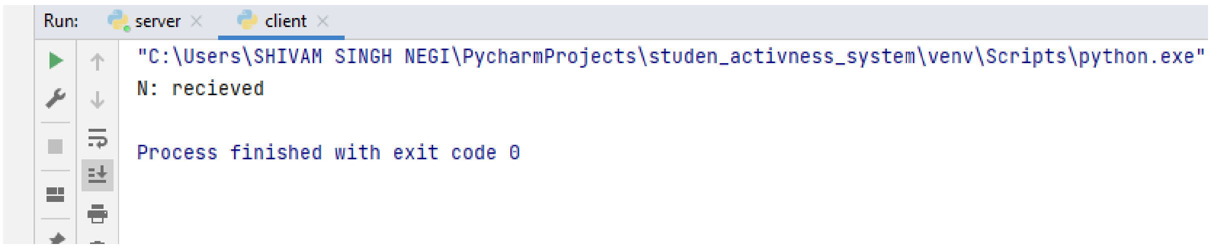
4.1.2 Data send from client to server



```
"C:\Users\SHIVAM SINGH NEGI\PycharmProjects\studen_activness_system\venv\Scripts\python.exe" "C:/Us
Socket Up and running with a connection from ('127.0.0.1', 62567)
S: 1912610,0.0,83.3,36.3,48.5
['1912610', '0.0', '83.3', '36.3', '48.5']
Written to file
```

Fig 4.2: Server side

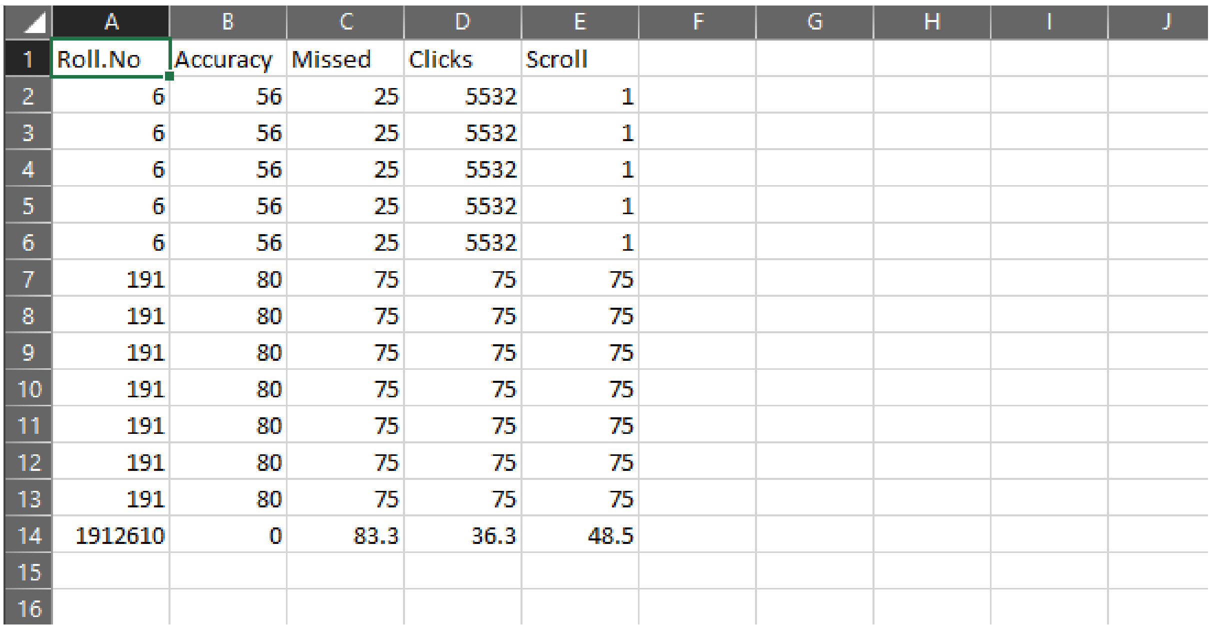
Figure 4.2 shows data collected at the client end being sent to the server in the form of string . The server accepts the sting and splits it using " , " as a separator and store it in a csv file



```
"C:\Users\SHIVAM SINGH NEGI\PycharmProjects\studen_activness_system\venv\Scripts\python.exe"
N: recieved

Process finished with exit code 0
```

Fig 4.3: Client Side

Fig 4.3 shows that when the receiver receives the data from the client it sends an acknowledgement back in form of a string “ received” .



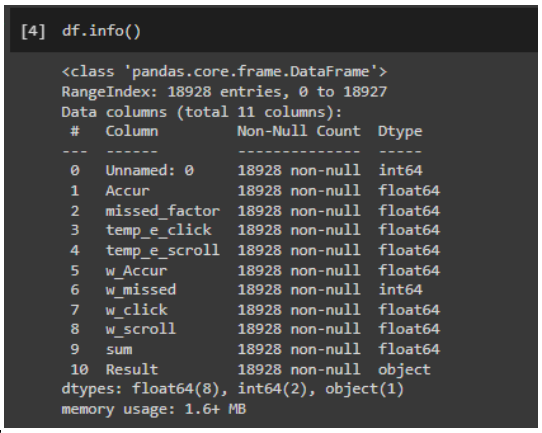
	A	B	C	D	E	F	G	H	I	J
1	Roll.No	Accuracy	Missed	Clicks	Scroll					
2	6	56	25	5532	1					
3	6	56	25	5532	1					
4	6	56	25	5532	1					
5	6	56	25	5532	1					
6	6	56	25	5532	1					
7	191	80	75	75	75					
8	191	80	75	75	75					
9	191	80	75	75	75					
10	191	80	75	75	75					
11	191	80	75	75	75					
12	191	80	75	75	75					
13	191	80	75	75	75					
14	1912610	0	83.3	36.3	48.5					
15										
16										

Fig 4.4: File stored in server

Fig 4.4 shows the file which is stored at the server end and the data which is sent from the client is stored here.

4.1.3 Dataset

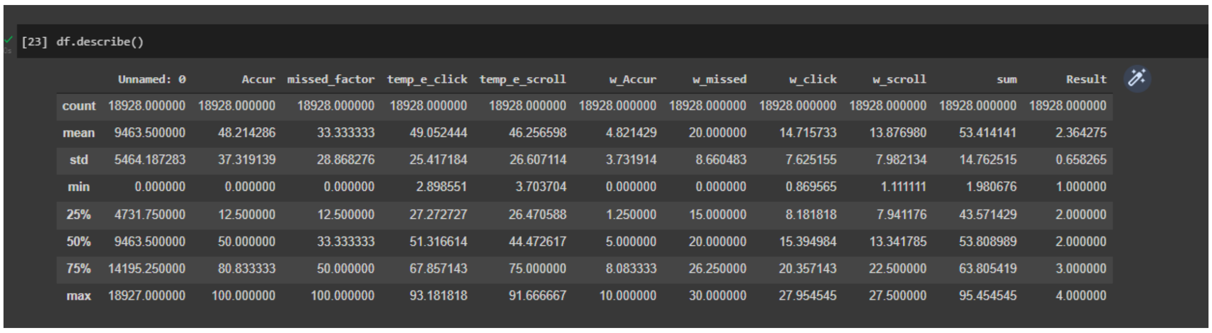
a) Description of dataset



```
[4]  df.info()

     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 18928 entries, 0 to 18927
     Data columns (total 11 columns):
      #   Column         Non-Null Count  Dtype
     ---  ------         --------------  -----
      0   Unnamed: 0     18928 non-null  int64
      1   Accur          18928 non-null  float64
      2   missed_factor  18928 non-null  float64
      3   temp_e_click   18928 non-null  float64
      4   temp_e_scroll  18928 non-null  float64
      5   w_Accur        18928 non-null  float64
      6   w_missed       18928 non-null  int64
      7   w_click        18928 non-null  float64
      8   w_scroll       18928 non-null  float64
      9   sum            18928 non-null  float64
      10  Result         18928 non-null  object
     dtypes: float64(8), int64(2), object(1)
     memory usage: 1.6+ MB
```

Fig 4.5: Attributes in dataset

In Fig 4.5 all the final attributes of the final dataset and their type are shown .



```
[23] df.describe()

        Unnamed: 0        Accur  missed_factor  temp_e_click  temp_e_scroll      w_Accur     w_missed      w_click     w_scroll          sum       Result
count  18928.000000  18928.000000  18928.000000  18928.000000  18928.000000  18928.000000  18928.000000  18928.000000  18928.000000  18928.000000  18928.000000
mean    9463.500000     48.214286     33.333333     49.052444     46.256598      4.821429     20.000000     14.715733     13.876980     53.414141      2.364275
std     5464.187283     37.319139     28.868276     25.417184     26.607114      3.731914      8.660483      7.625155      7.982134     14.762515      0.658265
min        0.000000      0.000000      0.000000      2.898551      3.703704      0.000000      0.000000      0.869565      1.111111      1.980676      1.000000
25%     4731.750000     12.500000     12.500000     27.272727     26.470588      1.250000     15.000000      8.181818      7.941176     43.571429      2.000000
50%     9463.500000     50.000000     33.333333     51.316614     44.472617      5.000000     20.000000     15.394984     13.341785     53.808989      2.000000
75%    14195.250000     80.833333     50.000000     67.857143     75.000000      8.083333     26.250000     20.357143     22.500000     63.805419      3.000000
max    18927.000000    100.000000    100.000000     93.181818     91.666667     10.000000     30.000000     27.954545     27.500000     95.454545      4.000000
```

Fig 4.6: Description of dataset

In Fig 4.6, the description of the dataset is shown which describes the dataset with the help of mean, mode , median of each attribute present in the dataset.

b) Distribution Graphs of various Attributes of Dataset

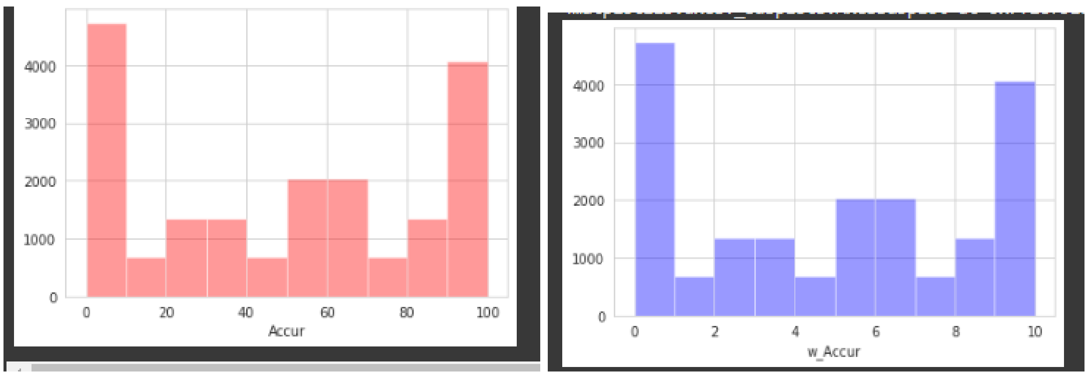


Fig 4.7 : Distribution plot of normal accuracy vs weighted accuracy

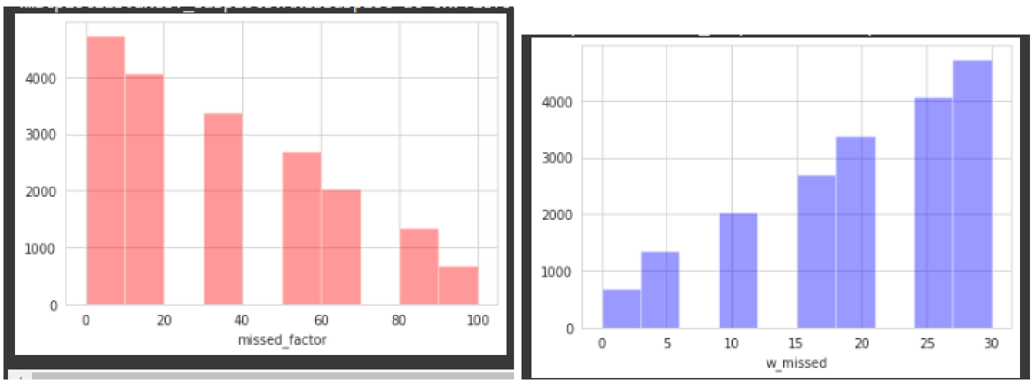


Fig 4.8: Distribution plot of normal missed factor vs weighted attempted factor

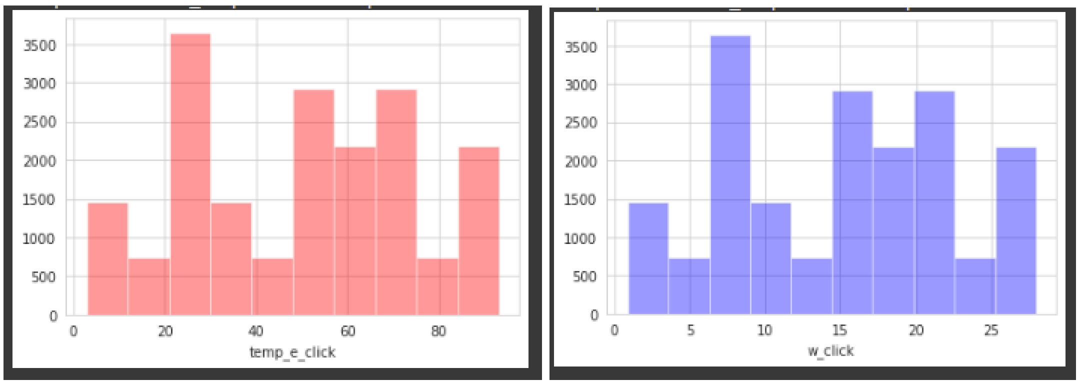


Fig 4.9: Distribution plot of normal clicks vs weighted clicks

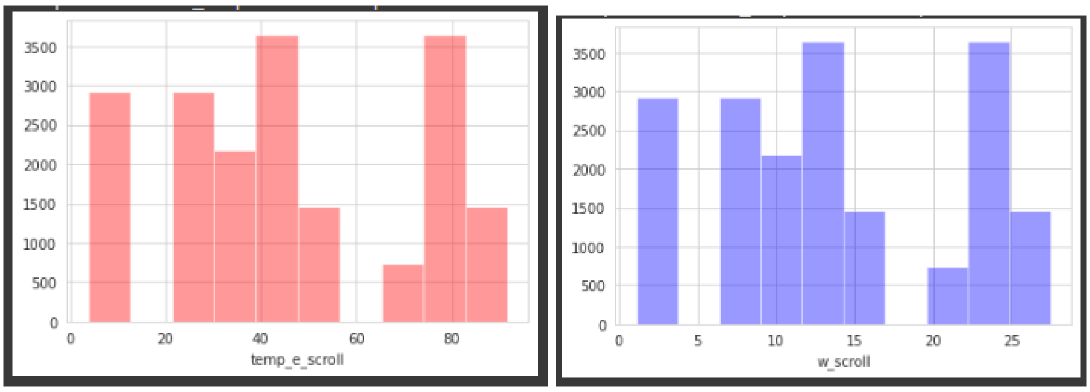
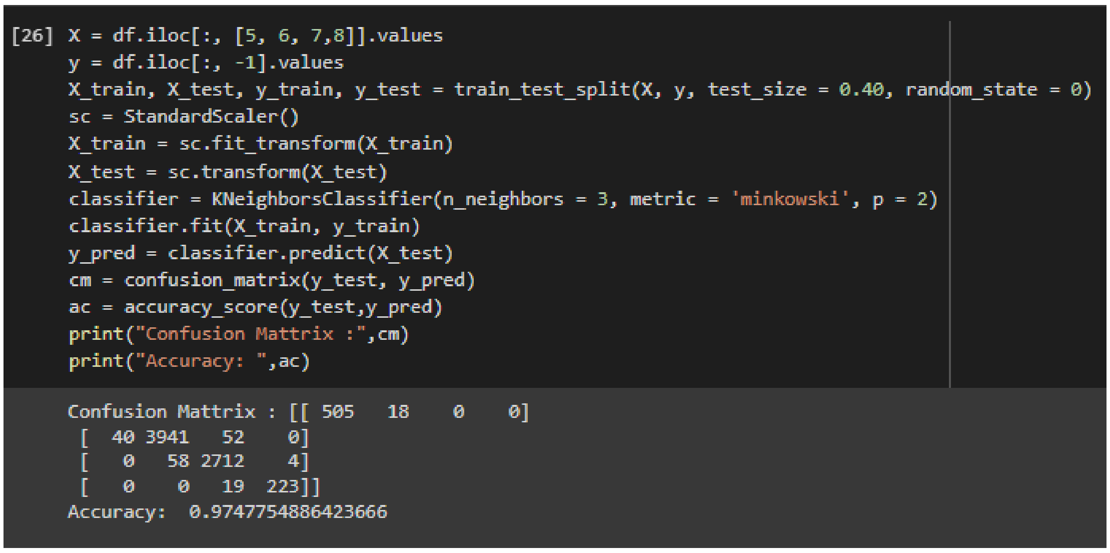


Fig4.10: Distribution plot of normal Scrolls vs weighted Scrolls

c) Classification Model : KNN Algorithm



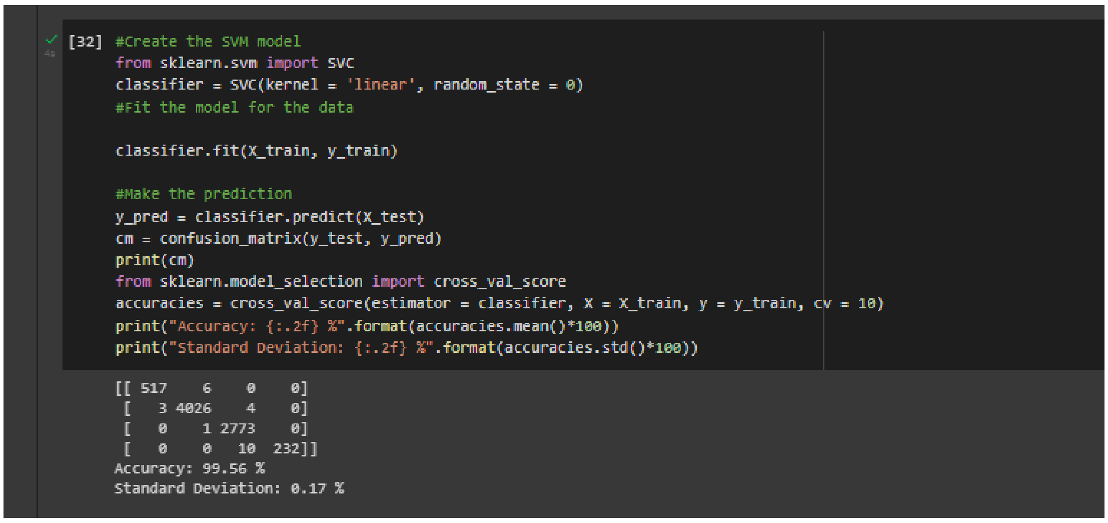
```
[26] X = df.iloc[:, [5, 6, 7,8]].values
     y = df.iloc[:, -1].values
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.40, random_state = 0)
     sc = StandardScaler()
     X_train = sc.fit_transform(X_train)
     X_test = sc.transform(X_test)
     classifier = KNeighborsClassifier(n_neighbors = 3, metric = 'minkowski', p = 2)
     classifier.fit(X_train, y_train)
     y_pred = classifier.predict(X_test)
     cm = confusion_matrix(y_test, y_pred)
     ac = accuracy_score(y_test,y_pred)
     print("Confusion Mattrix :",cm)
     print("Accuracy: ",ac)
```

```
     Confusion Mattrix : [[ 505   18    0    0]
      [  40 3941   52    0]
      [   0   58 2712    4]
      [   0    0   19  223]]
     Accuracy:  0.9747754886423666
```

Fig4.11: Implementation of KNN Algorithm

Fig 4.11 shows the pure implementation of KNN classification algorithm in which first of all the dataset is split into two parts and then the model is trained through X_train and y_predict is predicted for that. Ten the confusion matrix is crated and accuracy of model is calculated

d) Classification Model : SVM Algorithm



```
[32] #Create the SVM model
     from sklearn.svm import SVC
     classifier = SVC(kernel = 'linear', random_state = 0)
     #Fit the model for the data

     classifier.fit(X_train, y_train)

     #Make the prediction
     y_pred = classifier.predict(X_test)
     cm = confusion_matrix(y_test, y_pred)
     print(cm)
     from sklearn.model_selection import cross_val_score
     accuracies = cross_val_score(estimator = classifier, X = X_train, y = y_train, cv = 10)
     print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
     print("Standard Deviation: {:.2f} %".format(accuracies.std()*100))
```

```
     [[ 517    6    0    0]
      [   3 4026    4    0]
      [   0    1 2773    0]
      [   0    0   10  232]]
     Accuracy: 99.56 %
     Standard Deviation: 0.17 %
```

Fig 4.12: Implementation of SVM Algorithm

In Fig 4.12 the linear SVM classification algorithm is implemented and accuracy for the model is calculated for the same

4.2 Application

- It can be used as a monitoring system to keep track of students' activity during online classes without compromising their privacy.
- It can be used as a classifying system in which we can classify the student into four groups(Highly Active (Level 1), Active (Level 2),Less Active(Level 3), Least Active(Level 4)) based on their activeness during class.
- It can be used as an assessment system with the help of which the teacher can assign marks based on the mcq's and the activeness.

4.3 Limitation

- The machine learning classification model is not 100% accurate.
- The performance of the model is limited by the types and number of input devices used. For example, Keyboard and WebCam have not been considered for the model.

4.4 Future Work

- This feature can be added to online classroom platforms like Google Meet, Zoom meet etc as an extension.
- Recommendation System can be added so that interaction in class could be improved
- Some other

==

[1] https://www.geeksforgeeks.org/socket-programming-python/ has been used for implementing the client sever model

[2] https://realpython.com/python-sockets/ has been used for socket programming

[3] https://www.geeksforgeeks.org/k-nearest-neighbor-algorithm-in-python/ has been used for implementing KNN Algorithm

[4]
https://www.geeksforgeeks.org/classifying-data-using-support-vector-machinessvms-in-python/ has been used for implementing SVM Algorithm