# Problem statement

1. The system needs to calculate the average confidence score from 10 previous messages for each user and based on the score change models.
2. The system has to log the input message and the confidence score with the format specified.
3. The system must identify, calculate confidence score and log inputs separately for each user.

# Proposed solution

Cookies can be used to identify the visitor uniquely. Thus, it can be used to track each user’s visits even if they haven’t actively identified themselves, i.e., logged into the E-commerce store or web app. Since we can’t register users we opt to use cookies to identify users and log input messages and also calculate the confidence scores.

1. Users open the chat and we check if a cookie is present in the headers.
2. If there is a cookie we parse the header, take the cookie and send it to the model server in the payload.
3. If there is no cookie then we generate a cookie, send the cookie to the browser and also to the model server.
4. The model server takes the cookie from the payload and creates two files, one text file for the output and the other one csv file to store the confidence scores they will be stored in the input and confidences folder respectively(which will be created automatically when the model server starts in the run.py file.
5. Then the confidence score will be calculated and the input text will be formatted and logged in the output file.

To calculate the average I used a csv file to store previous values for the user then calculate and set the average value after the 10th message onward.

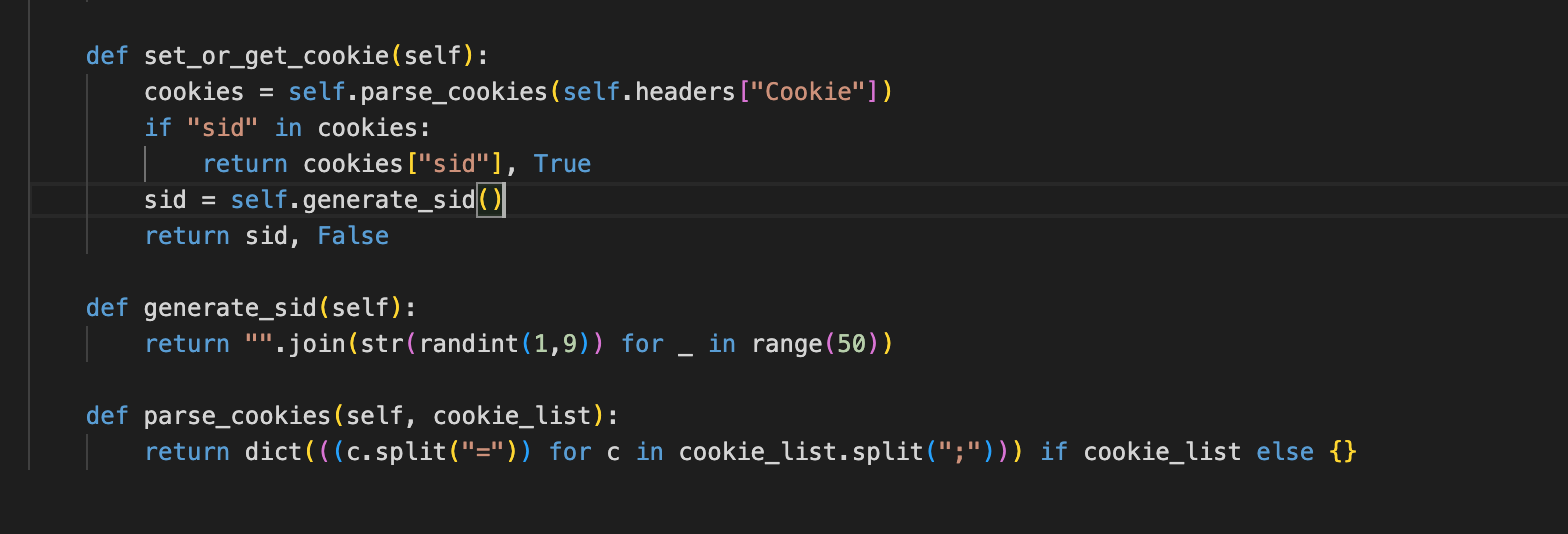
# Changes in the run.py file

Helper method to create the output and confidence directories if they are not present when the server starts.

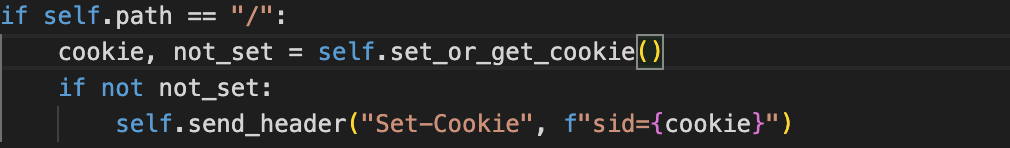


# Changes in the clients.py file

These are the helper methods used to generate the cookie, parse the cookie and set or create the cookie.

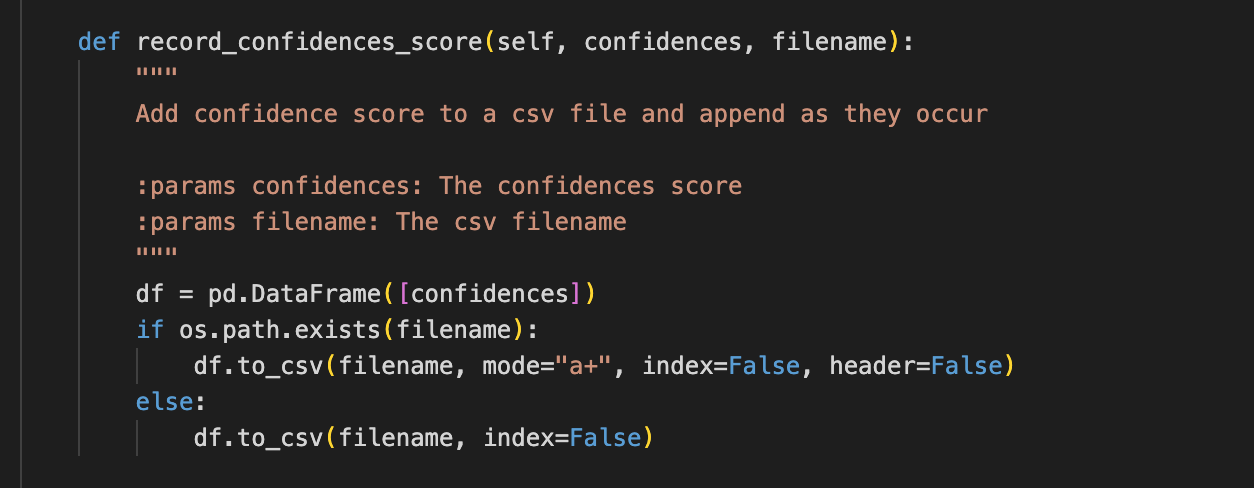


And we set the cookie like the code snippet below.

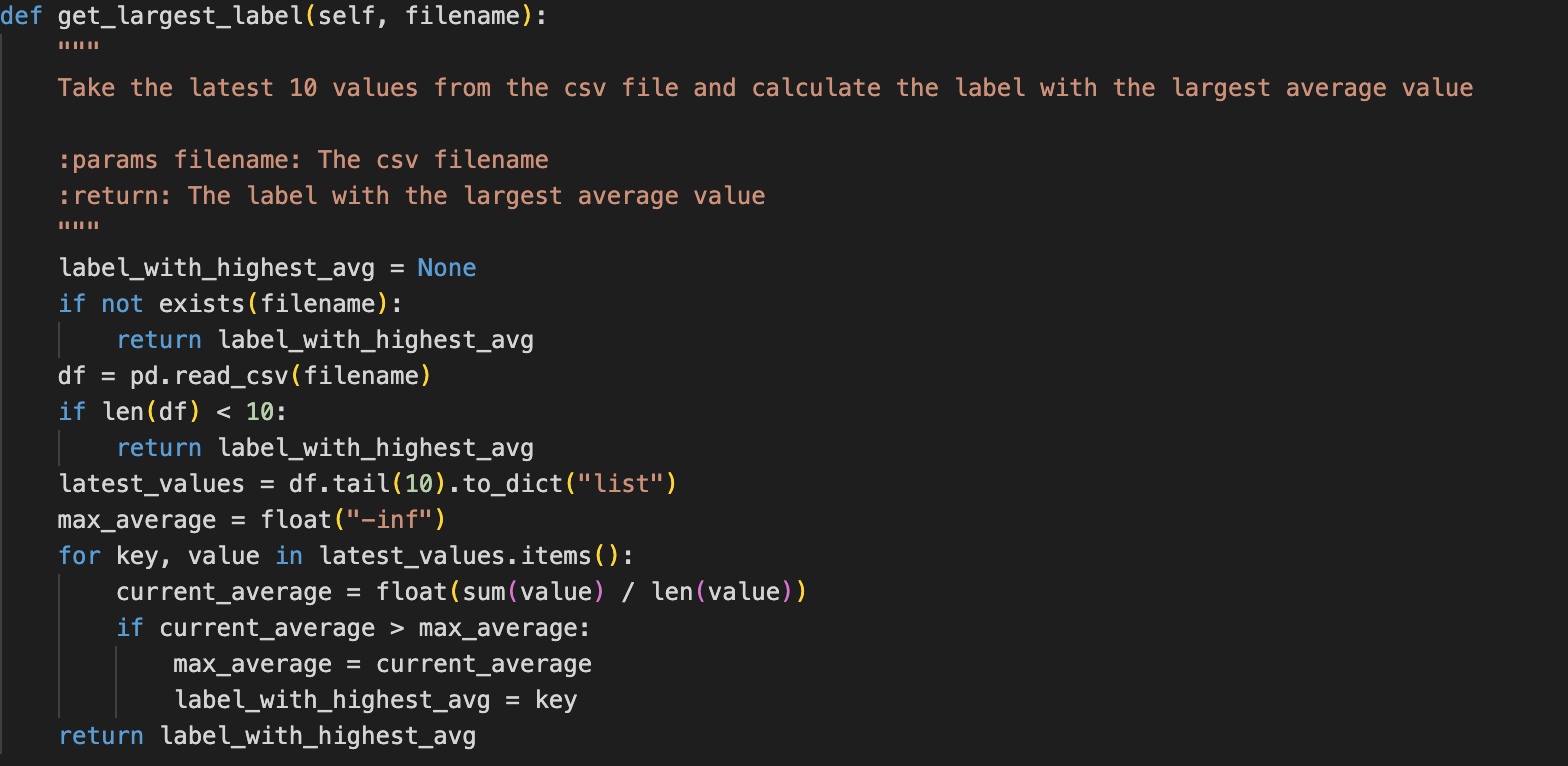


# Changes in the worlds.py file

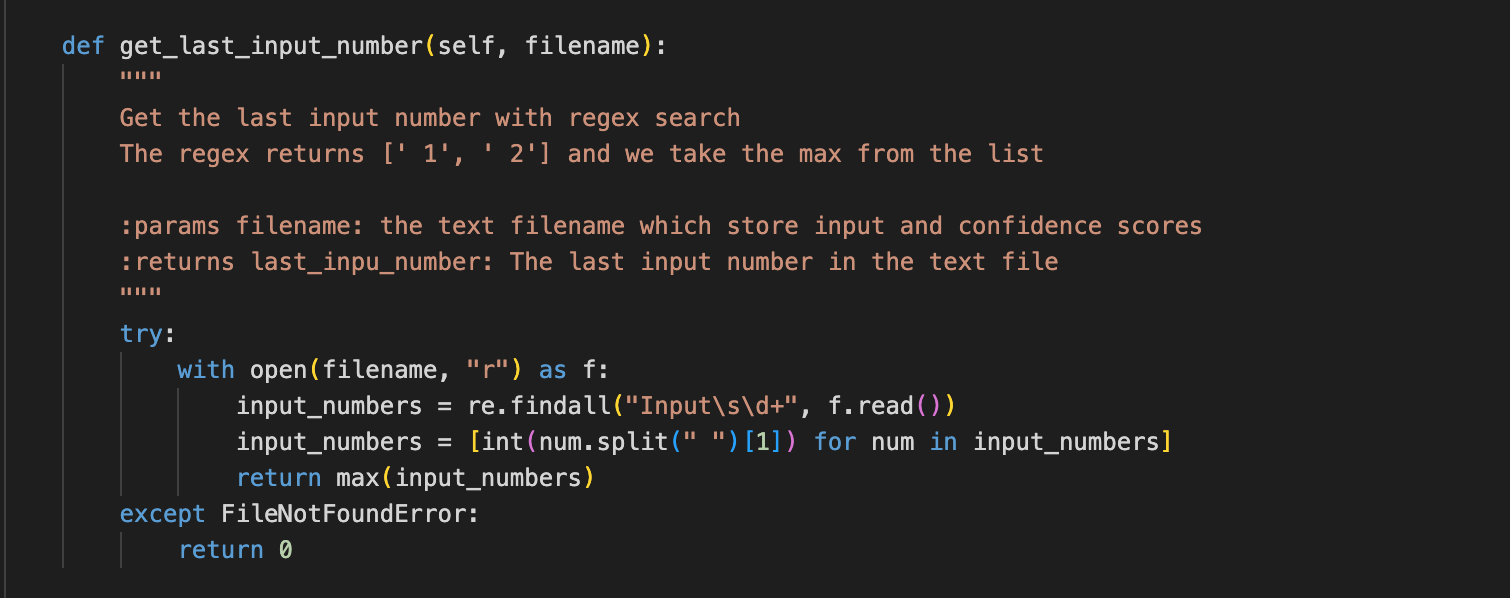
Helper method to record confidence score in csv file.



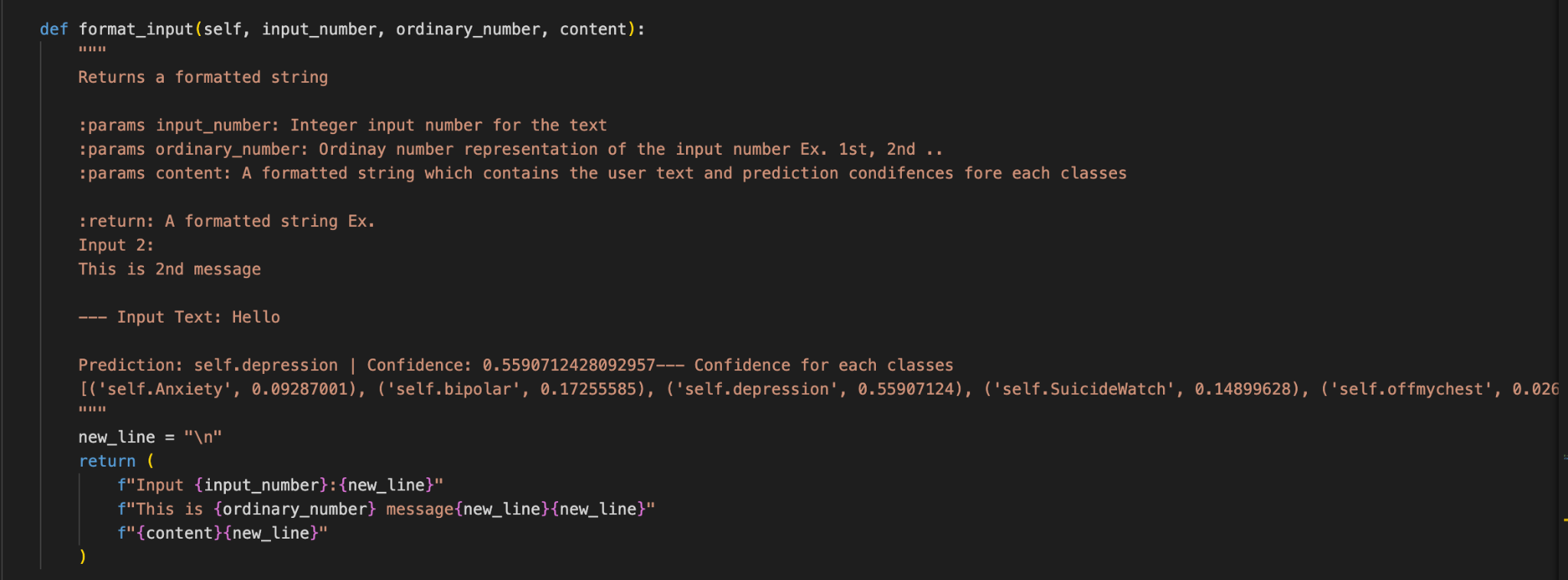
Helper method to get the largest label after calculating the average score for the last 10 messages.



This method helps to get the last input text number from the output file which uses regex to get the number.



Helper method to format the string to write to the output file.

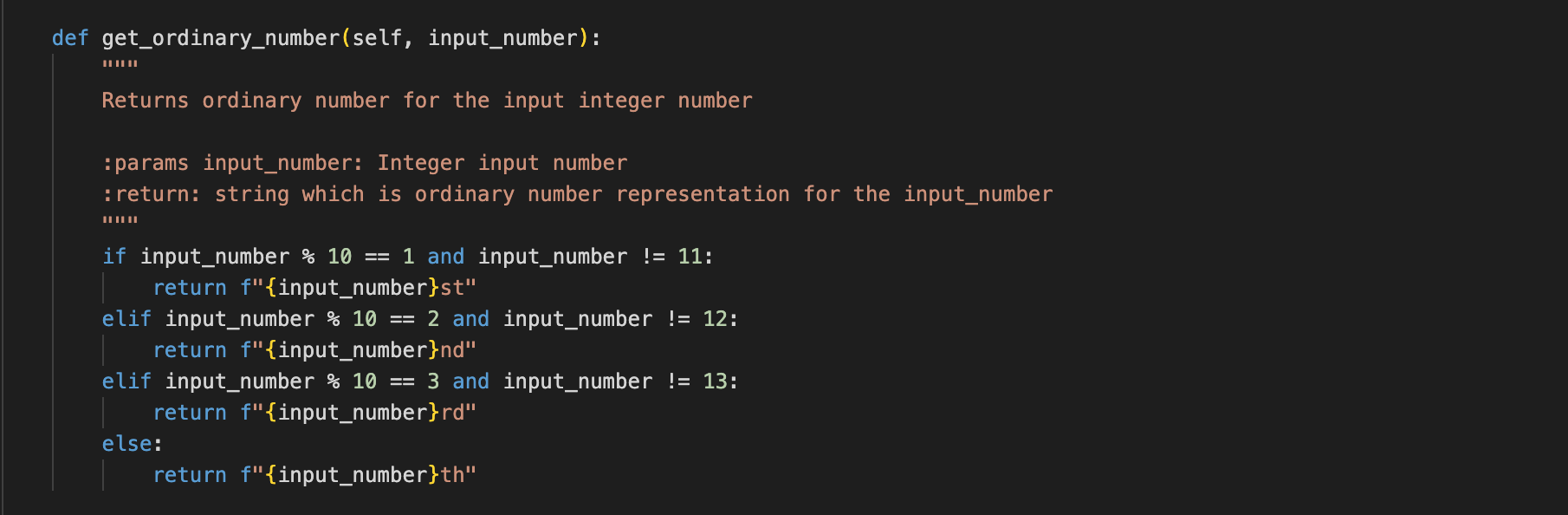


Helper method to change the number from integer to ordinary number for example

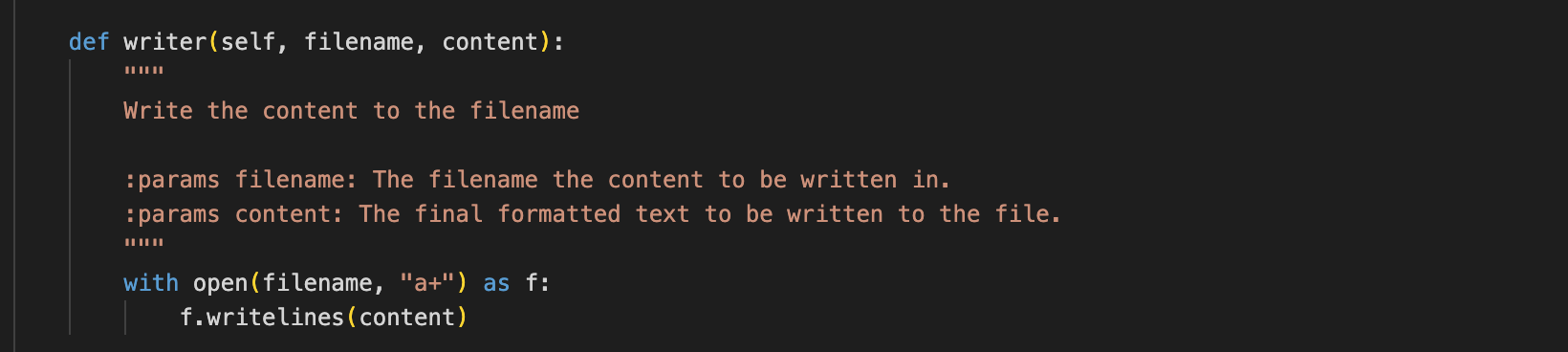
1- 1st

2- 2nd

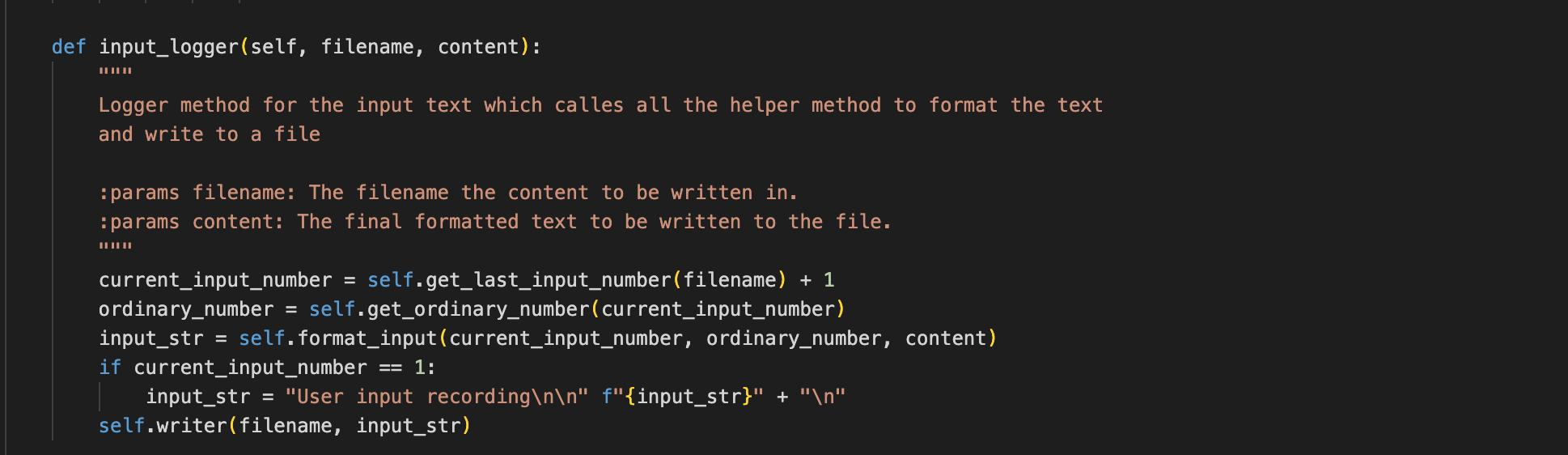
3- 3rd ..



Helper method to write to a file which doesn’t know what to write. Just writing the content to the filename this pure function does one thing well which helps to be used from different places and makes the code easily maintainable.



The main logger which calles all the other helper methods and write the content to the file using the previous writer method.



This is the code added to the parley method which created the file for the user based on the cookie passed in the payload then it prepares the content to be written to the file and calls the method which are mentioned above.

After the confidence score is calculated for the 10 input texts this method writed the confidence score with the average to the file using the writer method.



To make this work instead of using docker try creating a virtual environment that is much easier.

Go to the root directory

- create virtual env in the root directory  
python3 -m venv venv  
- activate the virtual directory  
source venv/bin/activate  
- the install the requirements (with this requirements.txt file)  
pip install -r requirements.txt --extra-index-url <https://download.pytorch.org/whl/cpu>  
- then run the model server on port 8080 as  
python3 [http://run.py](http://run.py/) --config-path ./config.yml --port 8080