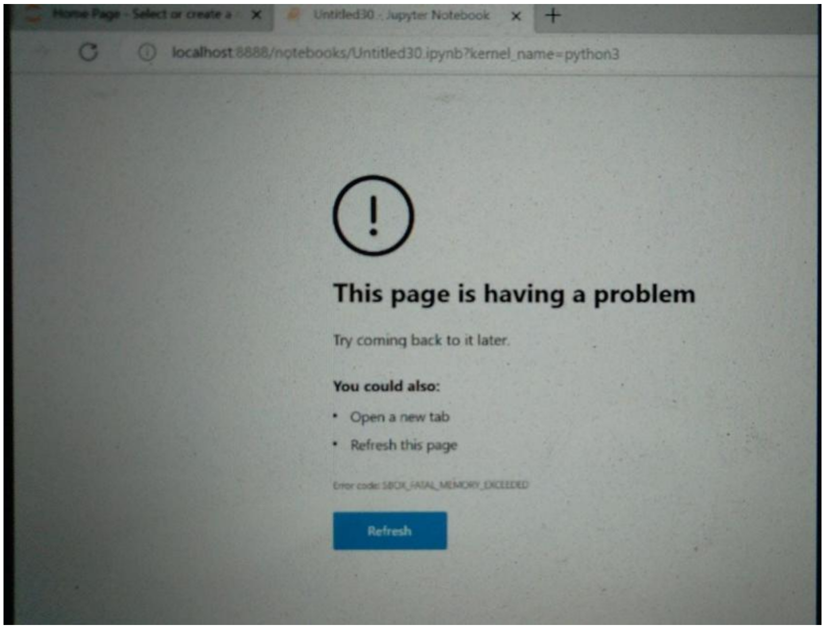
Team Name: **DREAMERS**
Team Lead: **Barisan Chatterjee**
Team Member: **Debapratim Sarkar**

Uniqueness of the approach selected:


Home Page - Select or create a
Untitled30 - Jupyter Notebook
localhost:8888/notebooks/Untitled30.ipynb?kernel_name=python3
Refresh

This page is having a problem

Try coming back to it later.

You could also:

- Open a new tab
- Refresh this page

Error code: SBOX_FATAL_MEMORY_EXCEEDED

As shown in the above screenshot, we were facing significant challenges in working with the Training data .**csv** file, with 80,000 rows. When I tried to handle the complete data at one go, Jupyter notebook in my laptop started returning this error: "Fatal Error – Memory Exceeded"
We took it as a blessing in disguise, and thought of giving a solution without a ML model, because at the end of the day, we were assigned with a classification assignment/challenge. **And we came up with a Python code which is currently classifying with an accuracy of 30%.**

Steps / Thoughts involved in the Plan of Action:

1. Initially we were provided with a sample file Sample.xlsx, which contained 10 rows with the text as well as their respective class classification (0-3)
2. **We analyzed and observed to decide that we will convert every hexadecimal text to decimal, and from each resulting decimal number we will take the summation of the digits, and also the average of the digits in each number. This is the primary logic which we used to classify the hexadecimal texts. (Refer code – CodeForInitialSample.py)**
3. Furthermore, we inferred on the conditions on the basis of the existing classification of the 10 hexadecimal texts in the sample file (Sample.xlsx).
4. Implemented the conditions decided in step 3, in our python program (**MAIN.py**), and accordingly our program started automatic classification of the hexadecimal texts in TrainingData.csv, with an accuracy precision of **30%**.
5. **Due to memory capacity challenges, we ran our python program on top 40,000 rows, and the output has been verified for the top 20 rows and the bottom 20 rows, out of those total top 40,000 rows**.