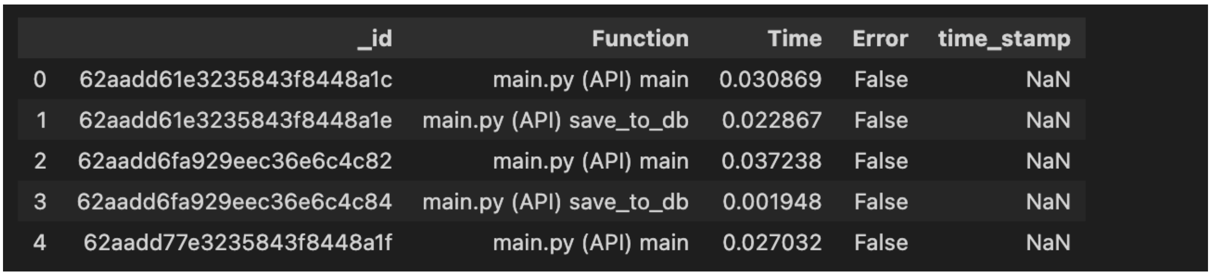
# Technical Interview:

Q:
Using the python library ‘**Streamlit’,** develop a web-app to visualise the ‘function’ column contents along with the ‘time_stamp’ and ‘Time’ columns provided in the dataset. We want the app to take in the number of rows we want to visualise. For example, 100 or 1000 or ‘n’ number of rows of the dataset. At the end you can provide a link for us to view the app you’ve deployed anywhere and as well as a GitHub repository for us to check the code you’ve implemented.

Some pointers:
Function column is like this in the dataset:



| | _id | Function | Time | Error | time_stamp |
|---|---|---|---|---|---|
| 0 | 62aadd61e3235843f8448a1c | main.py (API) main | 0.030869 | False | NaN |
| 1 | 62aadd61e3235843f8448a1e | main.py (API) save_to_db | 0.022867 | False | NaN |
| 2 | 62aadd6fa929eec36e6c4c82 | main.py (API) main | 0.037238 | False | NaN |
| 3 | 62aadd6fa929eec36e6c4c84 | main.py (API) save_to_db | 0.001948 | False | NaN |
| 4 | 62aadd77e3235843f8448a1f | main.py (API) main | 0.027032 | False | NaN |

The graphs for each of the functions (ex: main.py(API) main, main.py(API) save_to_db,etc.) should be displayed as a grid with y-axis being the ‘**Time**’ value and x-axis being the ‘**time_stamp**’

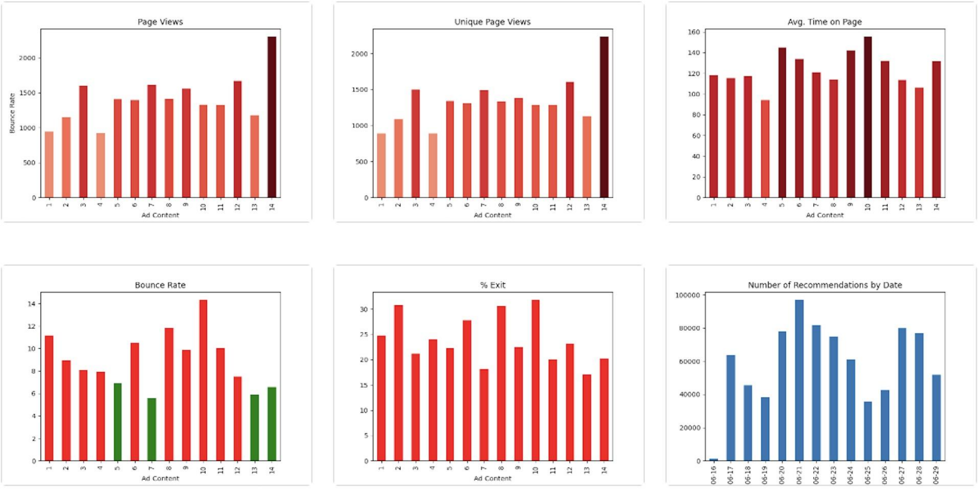
Use the existing data preprocessing methods to clean the data if necessary and if you find any NaN values in the time_stamp column, use the dataset where time_stamp is present.

Please document the entire process and include that in the repo that you’ll provide.

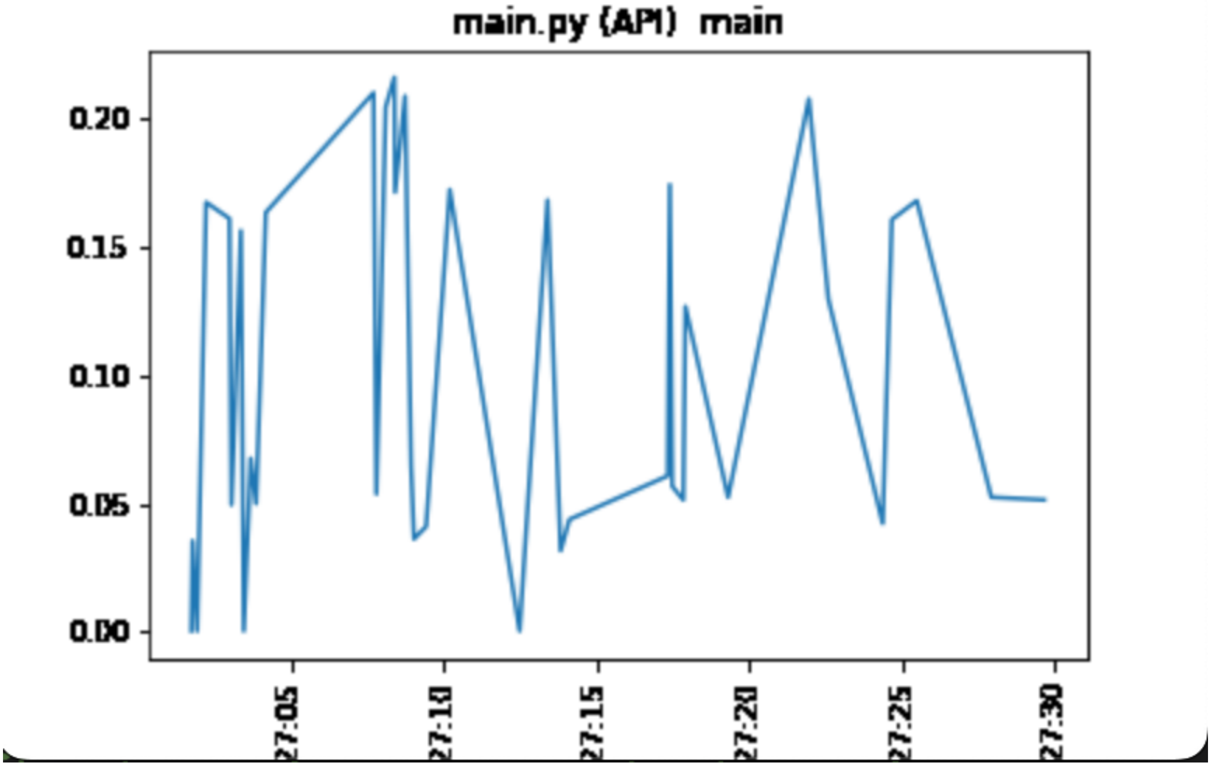
Time to finish this mini-project: Approximately 3-4 days.

**Deadline: 2th April 2023**

Example of the grid-based graph view:



Example of one function in the dataset:



2. I want you to utilise **‘Streamlit’** to create a user-friendly file upload interface and leverage FastAPI to process various text(s) in the excel file and return the count of the values of document ‘log_type’ and the ‘classification’ ’in a scalable and efficient manner? Moreover, what is the best approach to log each API request and persist the data in a database or CSV file that can be easily queried and analysed by other team members or stakeholders? Additionally, how can the team ensure that the logging process does not impact the performance or functionality of the web application?

Ex of API return data: {‘log_type_1’:21,’log_type_2’:19,’classification’:3’}

## Here are some of the theoretical questions we want you to write it in a document:

1. What is the curse of dimensionality? Explain how it can affect the performance of machine learning models. What techniques can be used to address this problem?

2. What is normalisation in database design? Why is it important? Give an example of a situation where normalisation might be necessary.

3. How do we run a python program in linux as a ‘systemctl’ service? Give us certain examples of such cases

4. With the dataset provided above, come up with faster and efficient techniques for fetching and processing the data from MongoDB to your local machine. Give us code examples what you’ve done and why’s the query faster (Code commenting)

5. Create a cron job of a python file that needs to be run for every 15th of the month at 3:45PM GMT. It should execute from the startup and store the output in a log file.

   File name: testing.py in folder: /home/ubuntu/NL/test

   Add the code snippet to the document as well.

**Good luck and happy coding!!**