Steps for function visualizer:

Step 1: Go through the csv file for data preprocessing. I expected there to be some missing values/ outliers. But I found the entire time\_stamp column was empty. To rectify that, I decided to fill it in with random values from the last 3 hours. Next, I used text input to get input from user regarding number of rows (num\_rows) to be visualized.

Step 2: Load the csv files as a Pandas dataframe (called df). And I used num\_of\_rows variable for the nrows argument in order to visualize the number of rows.

Step 3: Finally, I plotted the graph (time v/s time\_stamp) using the unique function names that I retrieved using the findUnique function.

Theoretical questions:

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.

1. As the name suggests, the curse of dimensionality occurs when we are working with high-dimensional datasets. As the number of features (dimensions) increases, the amount of data needed grows exponentially. Due to this, it can be extremely difficult to process the data in an efficient manner. Since the model becomes extremely complex, this can lead to overfitting of the data. Therefore, the model may not perform well on untrained data. Some techniques to overcome these obstacles are:

* Regularization (to put a constraint on the complexity of the ML model)
* Feature selection and dimensionality reduction (pick out most important features and reduce the dimensions of the dataset)
* Sampling (take a smaller sample of the dataset for training model)

1. Normalization is a method of data organization to reduce redundancies and dependency. It can be thought of as reducing a large table into smaller tables and defining a relationship b/w said small tables. It is important because:

* Reduces data redundancy
* Increases data integrity
* Improves the performance of the database (improving the efficiency of indexing)

A situation where normalization would be helpful is:

Let’s say a company has a table of employees containing info about all the employees in the company (employee ID, name, job title, salary, etc). this table will most likely contain redundancies (changing job titles/ salaries etc.)

To normalize this table, we can create sub-tables such as “job title”/ “department”. We can link the “employees” table to the “job titles” and “departments” so that each employee has a unique job title and department. Therefore, normalizing the dataset helped us avoid data inconsistency and reduce redundancy.

1. Step 1. Create a .service file in ‘/lib/systemd/system/’, say systemctl\_test.service. You can do that with the command: ‘sudo nano /lib/systemd/system/systemctl\_test.service’.

Thus, the path to the file will be ‘/lib/systemd/system/systemctl\_test.service’.

Step 2. To ensure that the service would work, we need to specify ExecStart and Restart under [Service] in systemctl\_test.service.

1. Run the command: ‘crontab -e”

Enter the command below to create the cron job:

45 15 15 \* \* /home/ubuntu/NL/test/testing.py >> /home/test.log 2>&1