**Background**

This Python script scraps data from the website of Madhya Pradesh State Load Despatch Centre (MP-SLDC). The functions of a SLDC are quite similar to the functions of Independent Systems Operator ((ISO) in the US. Under the provisions of Electricity Act 2003, SLDC of every state is vested with functions to ensure integrated operation of power system within the state. According to the Section 32 of Electricity Act 2003, SLDC should

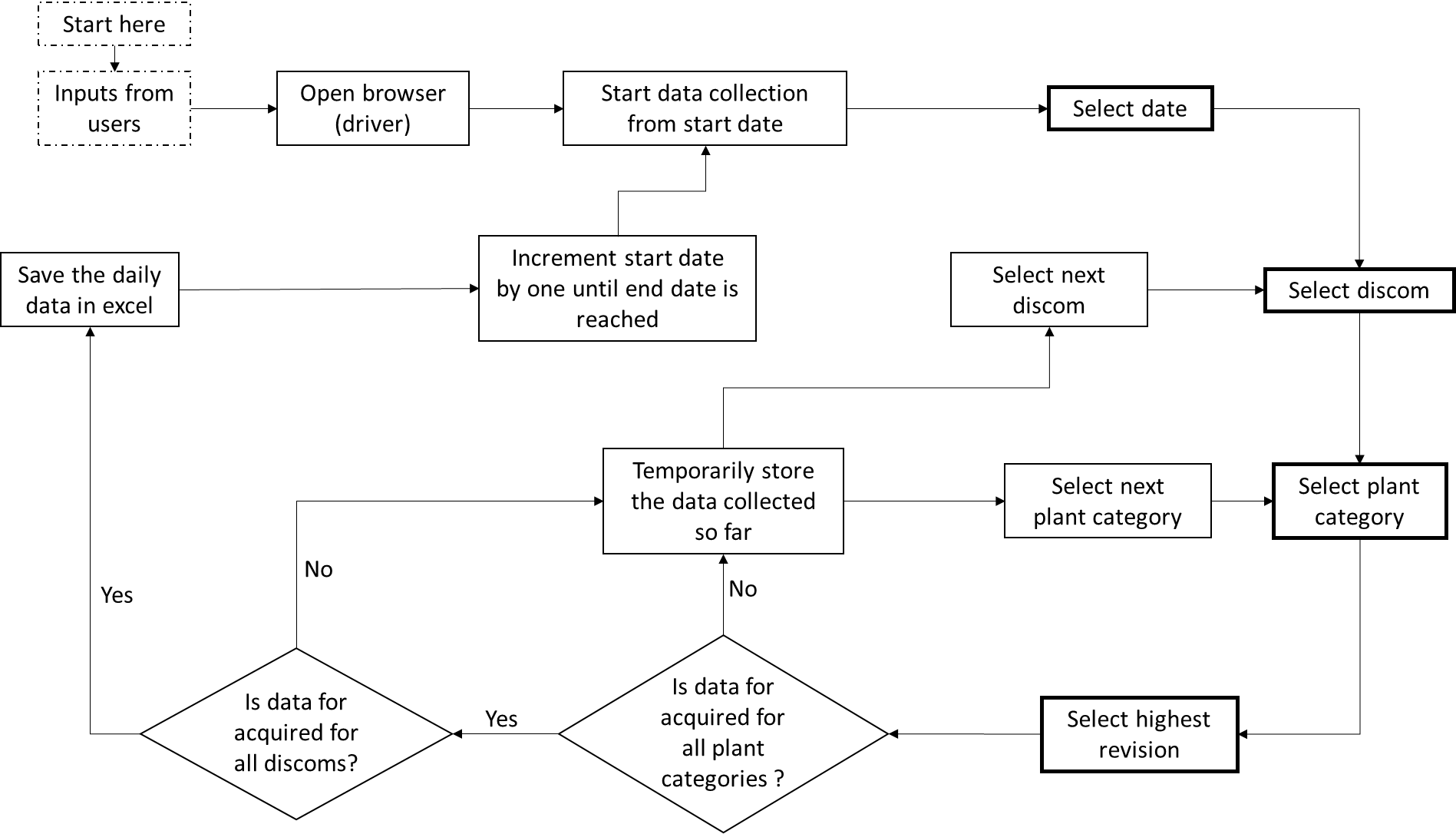
1. be responsible for optimum scheduling and despatch of electricity within a State, in accordance with the contracts entered into with the licensees or the generating companies operating in that State;
2. monitor grid operations;
3. keep accounts of the quantity of electricity transmitted through the State grid;
4. exercise supervision and control over the intra-state transmission system; and
5. be responsible for carrying out real time operations for grid control and despatch of electricity within the State through secure and economic operation of the State grid in accordance with the Grid Standards and the State Grid Code.

**Nature of Data**

The data represent block-wise drawl schedule of three power distribution companies (discoms) of Madhya Pradesh, namely, East Discom, West Discom, and Central Discom from various categories of power plants, namely, State Thermal (SSGS – T), State Hydro (SSGS – H), Independent Producers (IPP), and Inter-state power plants (ISGS). Each category has several generating stations of varying capacity, measured in MW. Each discom draws power from these plants. A time-block represents 15 mins time duration. Each day has 96 time blocks. Drawl schedule in a given time block refers to the power supplied, measured in MW, from a particular plant to particular discom during that time block.

**Architecture of the Script**

The basic architecture of the script is shown in the following figure.



**Salient features of script**

* The script uses functions for carry out many tasks. Thus, the script can be adapted for obtaining data from SLDCs of other states as well. The script can also be used to scrap injection schedule, which is relatively simpler.
* I have coded self-adjusting sleep time to optimize speed of data scraping. Whenever the data of 10 days are acquired, the sleep time will reduce to increase scraping speed, but if faster scraping results in frequent acquisition errors, the sleep time will also increase automatically to reduce the error. Based on frequent testing, it was observed if scraping was done quickly, then the response of the website was ‘No values found’.

**Potential Areas of Improvement**

This is my first independent Python project I did for self-learning. The script accomplishes data scraping while being robust to cases when StaleObjectException error occurs, webpage does not load entirely, and when no values are found on the website, as it happens many times.

However, I am sure there is still lot of room for improvements. From my side, as I learn, I will continue to improve the script w.r.t size of code, how fast it scraps data, its robustness to unforeseen errors, and memory requirements. I also request you to feel free to suggest any changes that will not only improve the code, but will also help me to learn more.

If possible, I request you to **look-into one issue in particular**. Currently, the speed of data scraping is very slow, as I have explicitly used ‘sleep’ feature multiple times to ensure that a webpage is loaded completely before scraping the data. I tried using ‘wait.driver’ feature, but it did not seem to work in this case. The website design is asynchronous. It requires four input parameters before the data is loaded, but the inputs must be given sequentially. For example, once you choose a date, then you must ‘submit’ and let the page load before you can select (from a dropdown menu) a discom. Once you select a discom, then the page loads automatically before you can select plant category and revision number. This makes process very slow. Whenever I tried increasing the speed, the data scraping went ‘out of sequence’ and the same data got acquired multiple times.

**Utility of the Script and Data**

The script may be useful for big-data learners and to some extent researchers who wish to apply data science in the field of energy management and require large amounts of readily available data. Descriptive statistics can be applied for analysing how the power demand of discoms vary over a period and how different categories of plants provide meet the demand. The findings may reveal highest and lower power demand over course of a day, types of plants i.e. thermal, hydro etc. providing the power, and carbon emissions associated with electricity generation.