Week 7 Assignment – AutoMPG Enhancements

Overview

In this assignment, you enhance your code from Week 6.

Please follow the steps outlined below.

Preparation

1. Copy the data files from the Week 6 assignment.
2. Copy **test_autompg.py** from the Week 6 assignment and rename it **test_autompg2.py**.
3. Copy **autompg.py** from the Week 6 assignment and rename it **autompg2.py**.
4. Adjust **test_autompg2.py** so that it tests the code in **autompg2.py**, and verify that it executes successfully.

Step 1: Configure logging module

In **autompg2.py** configure the **logging** module as shown in the lecture so that it logs at the DEBUG level into file **autompg2.log** and to the console at level INFO. Go back through the code in **autompg2.py** and add logging calls at appropriate places. As new code is written in this assignment, add logging calls as you go.

Step 2: Add sorting to AutoMPGData class

Enhance the AutoMPGData class by adding the following methods:

- **sort_by_default** – This method should use the **list.sort** method to sort the data list in place. By default, **list.sort** will use the sorting implied by the less-than operator already implemented on the AutoMPG class. After this method is run, the list will be sorted by **make, model, year**, and then **mpg**.
- **sort_by_year** – This method will use **list.sort** to sort the list in place but will override the default sorting by specifying the **key=** keyword parameter to **list.sort** to ensure that sorting happens by **year, make, model, mpg**.
- **sort_by_mpg** – This method will use **list.sort** to sort the list in place but will override the default sorting so that the order is determined by **mpg, make, model, year**.

Step 3: Add capability to download Internet data

Enhance the AutoMPGData class by adding the **_get_data** method. Please follow these guidelines for this method:

- **_get_data** will be called by the **_load_data** method if the original data file (**auto-mpg.data.txt**) does not exist.

- This method should use the **requests** module to download the original data file from the UCI Machine Learning Repository and save it in the local file **auto-mpg.data.txt**.

Test this functionality by removing the local data file and executing **autompg2.py**. It should download the data and then work as before.

Step 4: Enhance command-line parsing

In **autompg2.py** use the **argparse** module to implement enhanced command-line parsing for this program. The program should support the usage shown below. The only “command” supported is “print”, which will iterate over the elements of the **AutoMPGData** collection (as it is sorted) and print each one. The options to the “sort order” are “year”, “mpg”, and “default”, and the result of these options is to call the corresponding “sort_by_XXX” method on the **AutoMPGData** object before the data are printed.

```
usage: autompg2.py [-h] [-s <sort order>] <command>

analyze Auto MPG data set

positional arguments:
  <command>             command to execute

optional arguments:
  -h, --help            show this help message and exit
  -s <sort order>, --sort <sort order>
```

Upload

Please put **autompg2.py** into a ZIP file.