

DAV 5400 Fall 2019 Week 2 Assignment (30 Points)

A crucial aspect of working effectively with lists in Python is understanding how Python applies its internal indexing structure to lists. For this assignment you will be creating, searching, and sorting lists you will create from the contents of a small CSV file using some of the concepts we are learning about this week.

Start by downloading the **cars-sample35.txt** file to your local environment.

Then, open a new Jupyter Notebook and copy in the following small Python code snippet:

```
import csv
# be sure to update the path below to reflect your own environment!!
# also be sure that the code is properly indented after you paste it!

with open('yourpath/cars-sample35.txt') as csvfile:
    readCSV = csv.reader(csvfile)
    for row in readCSV:
        # print each row as read by the csv.reader function
        print(row)
```

This code will read the contents of a CSV file you specify in a line-by-line manner and print the contents of each line in the form of a Python list. Each item within that list will contain the corresponding element of the comma-separated line within the CSV file from which it came.

Once you've run the code snippet, you will see that each line of the file contains seven distinct values. For example, a print of the result of reading the first line will show the following:

```
['high', 'high', '2', '4', 'med', 'low', 'unacc']
```

These seven distinct values represent attributes of a single type of automobile. Specifically we have the following:

- Price
- Maintenance cost
- Number of doors
- Number of passengers
- Luggage capacity
- Safety rating
- Classification of vehicle

Your first task is to read the file again and extract these seven attributes from each line of the file and create seven distinct lists comprised solely of the values you extracted for a given attribute. In other words, you should have lists of prices, maintenance costs, number of doors, etc. For example, the first five "luggage" values should be as follows:

```
['med', 'small', 'big', 'big', 'med']
```

One way to complete this task would be to use the code snippet from above as a starting point: you could replace the "print(row)" statement with whatever Python code you feel is necessary to create the seven required lists.

Your second task is to find the list index values of each automobile having a price rating of "med". Create a new list with your result. *HINT: you can accomplish this task by searching the list of price values you created earlier.* Be sure to print your results.

Your third task is to find the "number of passengers" value for each auto having a "price" value of "med". Create a new list to store your findings and be sure to print your results.

Your fourth task is to find the index value for each automobile having a price value of "high" and a maintenance value that ***is not*** "low". Create a new list to store your findings and be sure to print your results.

Your fifth task is to find the index value for each auto having 2 doors and a luggage value of "big". Create a new list to store your findings and be sure to print your results.

Finally, create a new list containing the only the integer equivalents of the doors values. Keep in mind that the lists you have created thus far are composed solely of strings. If you find any values of '5more' in your list, convert them to a '5'. After converting the '5more' values to '5', convert all of the items in your list to their numeric equivalent and calculate the average number of doors across all 35 autos using whichever of Python's built in functions you require. Print your result.

Be sure to include some commentary explaining your approach to solving each of the individual problems. When you are finished, save the Jupyter Notebook containing your work and commentary and upload it to your online DAV5400 GitHub directory. Be sure to save your Notebook using the nomenclature we introduced last week, i.e., **first initial_last name_W2_assn** (e.g., J_Smith_W2_assn_).

As a reminder, this assignment is due no later than 11.59pm on Sunday Sep 8.