HW1 (Firebase)

DSCI 551, Fall 2022

100 points

Due: September 16, Friday, 11:59pm

Note: NO late submission will be accepted.

Consider a cars data set, cars.csv, which contains information about cars (car name, fuel type, number of doors, etc.), and prices. You may find more information about the data set here:

https://www.kaggle.com/code/goyalshalini93/car-price-prediction-linear-regression-rfe/notebook

Note that the data set you are given has different ids for the first and the last car as that in the original data set. Use the data set you are given in this homework. In general, do not assume that car id is one less than the row number.

1. [20 points] Write a Python script load.py that takes the CSV file cars.csv and loads it to your Firebase real time database. **Note: you need to load the entire data set to Firebase. Every car should be stored as a JSON object in Firebase.**

   Execution format: python3 load.py cars.csv

2. [20 points] Search by price range: write a Python script search_price.py that takes a range of price (in two arguments) and outputs IDs of cars in the range (inclusive). **Note: you need to create an index on car's price in the Rules of your Firebase database. If no cars are in the specified price range, then please return 'No cars found with the given range'.**

   Execution format (example): python3 search_price.py 15000 16000

Sample test Cases:
1. python search_price.py 15000 16000
IDs for the car price range are:
[6, 59, 112, …….]

2. python search_price.py 150 160

No cars found with the given range

3. [30 points] Create a keyword index: write a Python script create_index.py that creates a keyword index for keywords appearing in car name. The keywords in a car name are a list of words resulting from tokenizing the car name by punctuation characters **and white spaces**. For example, "alfa-romero giulia" has 3 tokens: alfa, romero, and giulia, while "dodge colt (sw)" has tokens: dodge, colt, and sw. For each unique keyword in the car names, **your index should store**

a list of IDs of cars whose name contains the keyword. You need to store the index in Firebase which will be used in the following search task. You should store index **separately** from the cars data set and should not overwrite the cars data set in Firebase. Note keywords are not case sensitive. So honda and Honda are considered to be identical.

Execute format: python3 create_index.py cars.csv

4. [30 points] Search cars by keyword: write a Python script search_car.py that finds IDs of cars using a list of keywords on car names. Note that search is case insensitive. Show the IDs of cars such that the cars whose name contains more keywords are shown first. You can break the ties arbitrarily. **Note the input may contain punctuation characters (e.g., entered by mistakes by users).**

   Execution format (example): python3 search_car.py “honda accord”

   Output should contain id’s with both honda and accord first, followed by cars whose name contains only honda or accord.

   If no cars contain any given keywords in their names, output “No cars found”. See example below.

Sample test Cases:

1.python search_car.py "honda accord"

IDs of the car are:
[41, 38, 36, 34, 43, 42 ……..]

2.python search_car.py "dodge coronet custom (sw)"
IDs of the car are:
[30, 28, 122, 29, 27, 25, 201, 195, ……]

Submission: Please submit a zip file by the deadline with the following content:

1. **The screenshot of layout of your database for tasks 1 and 3 (.png /.jpg)**
2. **The .py solution file for each task, that is, load.py, search_price.py, create_index.py, and search_car.py.**

Note: The given outputs are just samples and may not reflect correct answers.