**HW1**

**DSCI 551, Fall 2022**

**100 points**

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

1. [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

1. [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

1. [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”