NAP API Endpoints

Flask app to get NAP products related queries

Setup

- Install python3 or python2 : Ubuntu 18
  `sudo apt update` `sudo apt install python3`
- Install python3-pip
  `sudo apt install python3-pip`
- Install virtualenv for python2 (python3 have venv)
  `pip install virtualenv`

Virtual Enivronment

- Create python virtual envirnoment to contain project dependencies only
  `python3 -m venv flask_env`
- Now run the requirements.txt to install all flask dependencies
  `pip install -r requirements.txt`

Running Flask App

- `export FLASK_APP=app.py`
- `flask run`
  Running on http://127.0.0.1:5000/

Test (Using Postman tool)

Run in Postman

Press this button and open API request collection in postman and start playing with the APP.

Approach

First thing the program does is load the netaporter_gb.json file into the program memory as a list then the program converts that list into a normalized json (dataframe) which we store in a variable nap_dataframe.
Now we have the dataframe so next what I have done is creating a new column named **discounts** in dataframe containing discount for each NAP products because we need the discount a lot time in program so its better to calculate it once and store it as a column. Whenever,you hit the api endpoint with a POST request like `http://127.0.0.1:5000/greendeck/question1` the program firstly checks whether the posted json request is valid or not then it calls the required function such as in this case get_query_type1() passing it the filters object and this function then passes this filter to a common util filtering function which filters the dataframe based on the filters passed to it.
The program can be accessed or used using to type of API Endpoints

Type 1: `http://127.0.0.1:5000/greendeck/question1` (just change the question number and filters)
Type 2: `http://127.0.0.1:5000/greendeck/task` (just change the filters and hit ,the program will automatically call the required fucntion based on the query_type param in the json)
For the rest of approach look at the code as I have added detailed comments before a line of code wherever neccesary.

HEROKU API

- `https://greenapi.herokuapp.com/task`
- `https://greenapi.herokuapp.com/question1`
- `https://greenapi.herokuapp.com/question2`
- `https://greenapi.herokuapp.com/question3`
- `https://greenapi.herokuapp.com/question4`
- `https://greenapi.herokuapp.com/columns`

API Endpoints (Local)

- **GET Request** `http://127.0.0.1:5000/greendeck/columns`

  > This will return all the columns in the DataFrame used in this program.

- **Type 1 Query**
  POST Request `http://127.0.0.1:5000/greendeck/question1`

  > This will return NAP products ID having discount and brand.name filter in json request.
  > NAP products where discount is greater than n%

- **Type 2 Query**
  POST Request `http://127.0.0.1:5000/greendeck/question2`

  > This will return Count of NAP products from a particular brand and its average discount.

- **Type 3 Query**
  POST Request `http://127.0.0.1:5000/greendeck/question3`

  > This will return NAP products ID based competition filters in json request or previously used filters can also be used.
  > NAP products where they are selling at a price higher than any of the competition

- **Type 4 Query**
  POST Request `http://127.0.0.1:5000/greendeck/question4`

  > This will return NAP products ID based on competition with discount difference and other filters like used above.
  > NAP products where they are selling at a price n% higher than a competitor X.

- **Type All Query**
  POST Request `http://127.0.0.1:5000/greendeck/task`

> This is a all in one url endpoint which can be used in place of any of the 4 Post requests used above and return the same output.