**Data Scraping Assignment: TTC Customer Service Report**

Purpose:

Toronto Transit Commission (TTC) website updates Customer Service Report every day and displays the score card. The python script written is to fetch this score card table each day from the TTC website and save the data in a readable format in a database.

Prerequisites:

Environment used:

* Python 3.6.5
* IDE : PyCharm Community Edition
* GitHub link : <https://github.com/dhanyatha-harish/TTC_CS_report>
* PostgreSQL Database

Database Details:

* Database name : interview\_test
* Host : localhost
* User : postgres
* Password : Dhanyatha
* Table name : TTC\_CSR

Technical Summary:

External modules used:

* BeautifulSoup4 – To parse the HTML webpage to enable extraction of data easily
* Pandas – Data manipulation and analysis
* Psycopg2 – PostgreSQL adapter for python programming language
* Sqlalchemy – Python SQL tool kit and Object Relational Mapper

Steps:

1. Fetch the URL of the web page from which the data scraping is required
2. Use this URL to parse that web page through BeautifulSoup4
3. Use this object to find the table we are looking for in the HTML content using the class name of the table
4. Use a list to hold all the column names of the table which is fetched using beautifulsoup object
5. Use another list to hold all the values in the entire table using beautifulsoup object
6. Organize this list values which holds the entire table content, into each row value
7. We have header data in a list, each row data in another list, pass these as argument to create a DataFrame using Pandas
8. Use psycopg2 module to establish a connect with PostgreSQL data base, create a data base name , host name, user name and password
9. Create a table named “ttc\_csr” in “interview\_test” database
10. Push the DataFrame created using Pandas to this table using sqlalchemy module
11. We can run this every day to fetch Customer Service Report every day and append each day’s report to the same table we created the first time
12. This table from the database could be fetched as a DataFrame using Pandas and could be used further for Data Analysis
13. Try block is used where ever an exception is expected to make it easy for debugging purposes

Conclusion:

This script uses BeautifulSoup4 for parsing of webpage using HTML for easy extraction of data, Pandas to organize data in a DataFrame and use for data analysis or to store them in a database, sqlalchemy to easily push panda’s dataframe into PostgreSQL data base. The time complexity of this script is O(n). This is one of the ways, how data from web could be scraped and saved in a database.

Reference:

Below is a screenshot of how the DataFrame would look after retrieving it from Database which had multiple days of data in a table

