Documentation for HW1 on Firebase load, churn and tenure Methods

by Ruijie Rao 2022/01/25

Imported Libraries:

In [10]:

```
import pandas as pd
import json
import requests
```

Script 1: Load.py

Script Goal: load the rows for the above senior customers to your database.

Execution Command: python3 load.py

1.0 URL Config

In [13]:

```
db_url = "https://dsci551-173f1-default-rtdb.firebaseio.com/"
data_path = "WA_Fn-UseC_-Telco-Customer-Churn.csv"
```

1.1 Data Preparation

In [3]:

```
churn_raw_data = pd.read_csv(data_path)
```

In [4]:

```
churn_raw_data = churn_raw_data.set_index("customerID")
```

In [5]:

```
churn_raw_data.head()
```

Out[5]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	Inter
customerID								
7590-VHVEG	Female	0	Yes	No	1	No	No phone service	
5575-GNVDE	Male	0	No	No	34	Yes	No	
3668-QPYBK	Male	0	No	No	2	Yes	No	
7795-CFOCW	Male	0	No	No	45	No	No phone service	
9237-HQITU	Female	0	No	No	2	Yes	No	

In [6]:

```
churn_seniors_data = churn_raw_data[churn_raw_data["SeniorCitizen"]==1]
```

In [7]:

```
churn_seniors_data.head()
```

Out[7]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	Inter
customerID								
8779-QRDMV	Male	1	No	No	1	No	No phone service	
3841-NFECX	Female	1	Yes	No	71	Yes	Yes	
4929-XIHVW	Male	1	Yes	No	2	Yes	No	
3413-BMNZE	Male	1	No	No	1	Yes	No	
8012-SOUDQ	Female	1	No	No	43	Yes	Yes	

In [14]:

```
display(len(churn_seniors_data))
```

```
1142
```

Extract to Dict

In [11]:

```
churn_seniors_dict = churn_seniors_data.to_dict("index")
```

Example of dict structure:

In [12]:

```
display(churn_seniors_dict["8779-QRDMV"])
```

```
{'gender': 'Male',
 'SeniorCitizen': 1,
 'Partner': 'No',
 'Dependents': 'No',
 'tenure': 1,
 'PhoneService': 'No',
 'MultipleLines': 'No phone service',
 'InternetService': 'DSL',
 'OnlineSecurity': 'No',
 'OnlineBackup': 'No',
 'DeviceProtection': 'Yes',
 'TechSupport': 'No',
 'StreamingTV': 'No',
 'StreamingMovies': 'Yes',
 'Contract': 'Month-to-month',
 'PaperlessBilling': 'Yes',
 'PaymentMethod': 'Electronic check',
 'MonthlyCharges': 39.65,
 'TotalCharges': '39.65',
 'Churn': 'Yes'}
```

1.2 Data Upload

In [ ]:

```
url = db_url+"churn_data_by_ruijie.json"
payload = json.dumps(churn_seniors_dict)
response = requests.put(url,payload)
```

The only request sent by this script consists of *url* and *payload* two parts.

- **url** variable is a combination of the firebase db_url and the payload json name *churn_data_by_ruijie.json*.
- **payload** variable is the extracted dict from the imported data.

1.3 Request Feedback

- If the response from the Firebase server is 200(which is a pass), "Upload Complete!" will be printed.
- If the response from the Firebase server is not 200(which is an error), the response message in json will be printed.

In [ ]:

```
message = response.json()
if response.status_code == 200:
    print("Upload Complete!")
else:
    print(message)
```

Script 2: Churn.py

Script Goal: find the first k (senior) customers who has churned. Only need to return IDs of first k customers (ordered by their IDs).

Execution Command: python3 churn.py \<k>

In [15]:

```
def churn(k):
    #Config
    db_url = "https://dsci551-173f1-default-rtdb.firebaseio.com/"

    #Curl
    url = f'{db_url}churn_data_by_ruijie.json?orderBy="Churn"&equalTo="Yes"&limitToFi
    response = requests.get(url)
    filtered_data = response.json()
    result = list(filtered_data.keys())

    print(result)
```

The *url* variable adds a filter instead of a data payload.

- **orderBy="Churn"** which means the filtered column will be "Churn".
- **equalTo="Yes"** means the filtered data need to have "Yes" on the column "Churn"
- **limitToFirst={k}** means only want the first k number of data filtered.
- **print=pretty** means want the response to be formatted in specific way.

Script 3: Tenure.py

Script Goal: find out how many customers who have used the service for at least k months.

Execution Command: python3 tenure.py \<k>

In [16]:

```
def tenure(k):
    #Config
    db_url = "https://dsci551-173f1-default-rtdb.firebaseio.com/"

    #Curl
    url = f'{db_url}churn_data_by_ruijie.json?orderBy="tenure"&startAt={k}&print=prett
    response = requests.get(url)
    filtered_data = response.json()
    result = len(filtered_data.keys())
    print(result)
```

The *url* variable adds a filter instead of a data payload.

- **orderBy="tenure"** means the filtered column will be "tenure".
- **startAt={k}** means only want the data entries that are bigger or equal to k.
- **print=pretty** means want the response to be formatted in specific way.