Python Web Scraping

Abstract:

The theme for this exercise is to display the statistics about the countries in the world. I've collected data from various online websites and performed operations on them to extract specific. Information which I wanted to emphasize on. The data sets were taken using three different methods viz a pre-loaded csv file, scrapping the Wikipedia website and using a web api to display additional data about a country. All the data obtained is received in the raw form and is not simulated. It is compliant with real world factual data.

Conceptual Model:

The conceptual model contains the world and countries as the main **entities**. The countries and the world demonstrate **one-to-many relationship** as one world has many countries Here the name of the country forms the "**primary key"** which is common in all the 3 data sources and to which all the other attributes are defined.

- **CSV file**

  In the csv file all the attributes are related to the country with **one-to-many relationship.** One country has attributes of literacy, population, birthrate, death rate, etc.
- **Web Scrapper:**

  In the table scraped from Wikipedia, the type of constitution and head_of_state is related to the primary key "country name" by **one-to-one relationship**. A specific country has one type of constitution (republic or monarchy based).
- **Web Api:**

  In the data extracted from web api I have the country as the primary key. I extracted values of Longitude-Latitude, ISO-3 word country code and currency name. All these attributes are related to the country by **one-to-one relationship.**

Data Sources:

- **Csv file:** The csv dataset was obtained from www.techslides.com which included information about 227 countries in the world along with their statistics like country name, population, net migration, area, literacy(%), GDP, Climate, Birthrate, Death rate.
- **Web Scrapper:** We decided to scrape the www.Wikipedia.org website to get the data set about the constitution type for different countries. The web scrapper was given the task for scrapping the table tags on the web page by inspecting the html code of the Wikipedia web page.
- **Web Api:** The web api http://countryapi.gear.host/v1/Country/getCountries was used to parse for data and using json and urllib packages and pandas stored in datasets.

Data Processing:

- Using NumPy and pandas package the csv file was assigned to a variable “ver” and was analyzed. The **ver= pd.read_csv(“filename”)** function helped to assign the csv file to the variable defined. For cleaning the csv file the isnull().sum() checked for the number of null values in the entire data sets and displayed them. After finding the discrepancies the del command deleted those columns and the cleaned data columns reflecting the values were displayed.
- BeautifulSoup and Requests package was used for scrapping the web page. soup.find() method found the exact html tag of the table to be scrapped. An empty list was created for the attributes and content was appended to the list using a for-loop. Pandas library was used to create a data frame tags helped to display the data in a tabular format.
- The method with urllib.request.urlopen("http://countryapi.gear.host/v1/Country/getCountries") as url stored the data in the api in the variable url. Then using data = json.loads(url.read().decode()) I parsed the json file. Then using the dumps() method stores it as a json string. The it is stored in dataframe using a loop and append function. After checking for any errors or null values the data is stored in csv file

Auditing the data:

Data Cleaning:

- CSV File: The function “. isnull().sum()” extracted all the columns in the dataset which had null values.(Literacy, Net migration, Agriculture, Industry, Service all had null values. For database consistency these values were replaced with zero using “.fillna(0, inplace=True)” method. Then the updated csv file was saved to another csv file which will be later merged.
- Web Scrapping: During Web scrapping, in the table extracted from Wikipedia there were many values with “/n and /xa0” characters. These characters were replaced with empty characters (‘’) using the “.replace()” function call.
- Web api: For the web api there were ASCII value errors which were removed using data1.Country.replace({r'[^\x00-\x7F]+':''}, regex=True, inplace=True) expression.

Citations:

- The csv dataset : www.techslides.com/demos/country-capitals.csv. And reference from GitHub files of professor Nick Brown: https://github.com/nikbearbrown/INFO_6210/blob/master/Movie_DB_Example/TMDB_Movie_Data_Assignment_Example.ipynb
- For web scrapping, the GitHub files of professor Nick Brown were used as reference. https://github.com/nikbearbrown/INFO_6210/blob/master/Week_2/NBB_IMDB_Web_Scraper.ipynb

- The web api was taken from https://github.com/fabian7593/CountryAPI. The tutorial to read json data https://www.dataquest.io/blog/python-api-tutorial/ .