IBMCapstone

January 17, 2020

[ ]:
```
#import libraries
import numpy as np
import requests
import pandas as pd
import csv
!conda install beutifulsoup4
```

[ ]:
```
from bs4 import BeautifulSoup
import xml
```

1 Use the Notebook to build the code to scrape the following Wikipedia page, https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M, in order to obtain the data that is in the table of postal codes and to transform the data into a pandas datafram

[ ]:
```
#define URL to webscape
source = requests.get('https://en.wikipedia.org/wiki/
 ↪List_of_postal_codes_of_Canada:_M').text
soup = BeautifulSoup(source,'html')
#makes it east to read
print(soup.prettify())
```

[ ]:
```
#the table data we want is here
table = soup.find('table',{'class':'wikitable sortable'})
table
```

[ ]:
```
#looks as though the values of td are what we want so to find all listed as td
links = table.find_all('td')
links
```

[ ]:
```
#with this cleaner version we can finally start to scrape what we want
#creat lists of what we want from the table

postecode = []
```

```
borough = []
neighborhood = []
```

[ ]:

```
#scrapes values of td for only text and assigns them to there respective areas
for i in range(0, len(links), 3):
    postecode.append(links[i].find(text=True))
    borough.append(links[i+1].find(text=True))
    neighborhood.append(links[i+2].find(text=True).rstrip())
```

[ ]:

```
#puts it into a data frame
df_postalfields = pd.DataFrame(data=[postecode, borough, neighborhood]).
 ↪transpose()
df_postalfields.columns = ['Postecode', 'Borough', 'Neighborhood']
```

[ ]:

```
df_postalfields.head(20)
```

2 Only process the cells that have an assigned borough. Ignore cells with a borough that is Not assigned.

[ ]:

```
#Clean and remove postal codeds under borough that are labled not assigned

#make not assigned = missing value
df_postalfields['Borough'].replace('Not assigned', np.nan, inplace=True)
#remove missing values
df_postalfields.dropna(subset=['Borough'], axis=0, inplace=True)
df_postalfields.head(20)
```

3 More than one neighborhood can exist in one postal code area. For example, in the table on the Wikipedia page, you will notice that M5A is listed twice and has two neighborhoods: Harbourfront and Regent Park. These two rows will be combined into one row with the neighborhoods separated with a comma as shown in row 11 in the above table.

[ ]:

```
#combine similar neighborhoods with poste codes using comma

#Saves postecode and borough and shows combined neighborhoods in new data frame
df_pf = df_postalfields.groupby(['Postecode','Borough'])['Neighborhood'].
 ↪apply(','.join).reset_index()

#new columns for data frame
df_pf.columns=['Postecode','Borough','Neighborhood']
df_pf
```

4 If a cell has a borough but a Not assigned neighborhood, then the neighborhood will be the same as the borough. So for the 9th cell in the table on the Wikipedia page, the value of the Borough and the Neighborhood columns will be Queen's Park.

[ ]:
```
#Changes values in Neighborhood from not assigned to value listed in borough
df_pf['Neighborhood'].replace('Not assigned', 'Borough', inplace=True)
df_pf
```

5 In the last cell of your notebook, use the .shape method to print the number of rows of your dataframe.

[ ]:
```
df_pf.shape
```

[ ]:
```

```