

Capstone Project - The Battle of Neighborhoods - Taichung version

This project is a part of IBM data science; you will find in this post an overview of my final capstone project.

In this assignment, I will go through the problem description (**Introduction/Business Problem**), data set preparation and final to analysis and overview these data step by step. Detailed code is given in Github and link can be found at the end of the post.

1 Introduction/Business Problem

1.1 Background

Taiwan is a famous nation in the world. It was very small (36,197 km²) but with more than 23,600,000 population. Taiwan with the high metropolitan GDP in the world and it belonged a developed country.

There was COVID-19 spread in the world but it seems not impact to Taiwan. Here with the best epidemic prevention concept and medical technology so the infection rate and mortality rate is the lowest in the world.

With estimated 11,840,000 overseas visitors every year. It's famous with delicious foods and beautiful landscapes. Many travelers needed to know some leisure and entertainment places when they first visit to Taiwan.

These characteristics also attract many immigrations come to Taiwan.

Taichung is a second big city in Taiwan, its' urban scale was smaller than Taipei only. Taichung retained the advantage of Taipei like: convenient transportation/convenient life/good job opportunity/medical system...etc. Many immigrations will choose settle down in Taichung.

1.2 Business Problem

Taichung with 29 different districts, some districts are commercial and some are convenient and some are leisure. These districts are so difference!!!

In this article will help immigrations to understand and overview the Taichung city. Let them to choose the district which they want to live or travel.

2 Data and Data Preparation

2.1 Data Clean and Preprocess:

2.1.1 Define CSV Process Class

```
In [4]: class CSVprocess(object):

    def __init__(self):
        #self.address = "C:\\Users\\Brian\\Desktop\\ibm_coursea\\Coursera_Capstone\\week4\\"
        self.address = "C:\\Users\\brian\\Desktop\\Coursera_Capstone\\week4\\"

    def load_from_csv(self, filename):
        df = pd.read_csv(self.address + filename, encoding='UTF-8')
        return df

    def write_to_csv(self, filename, dictionary):
        column_list = []

        for key, value in dictionary.items():
            column_list.append(key)

        df = pd.DataFrame(dictionary, columns=column_list)
        df.to_csv(self.address + filename, index=False, encoding="utf_8_sig")
```

2.1.2 Define Json Process Class

```
In [5]: class JsonProcess(object):

    def __init__(self):
        #self.address = "C:\\Users\\Brian\\Desktop\\ibm_coursea\\Coursera_Capstone\\week4\\"
        self.address = "C:\\Users\\brian\\Desktop\\Coursera_Capstone\\week4\\"

    def load_from_json(self, filename):
        json_file = open(self.address + filename, "r", encoding="utf-8")
        j = json_file.read()
        json_file.close()
        dict_data = json.loads(j)
        return dict_data

    def dictionary_to_json(self, dictionary):
        json_data = json.dumps(dictionary)
        return json_data
```

2.1.3 Define Get Data from Web Class

```
In [6]: class GetDataFromWeb(object):

    def get_taichung_info(self):
        response = requests.get('https://zh.wikipedia.org/wiki/臺中市#人口')
        content = response.content
        df = pd.read_html(content, encoding='utf-8')
        taichung_info_df = df[11]

        return taichung_info_df
```

2.1.4 Define DataFrame Process Class

```
In [7]: class DataProcess(object):

    def __init__(self):
        pass

    def merge_dataframe(self, source_df, target_df, field_name):
        df = pd.merge(source_df, target_df, on=field_name)
        return df

    def drop_rename_dataframe(self, df, row_drop_list, column_drop_list, rename_list):
        df = df.drop(row_drop_list)
        df = df.drop(columns=column_drop_list)
        for item in rename_list:
            for key, value in item.items():
                df = df.rename(columns={key: value})

        return df
```

2.2 Taichung wiki:

Data source: <https://zh.wikipedia.org/wiki/臺中市#人口>

Description: This data set included 29 districts in Taichung city. There are some fields we need like: Chinese Name, Area, Density and Postal Code.

```
In [61]: row_drop_list = [29]
column_drop_list = ["下轄里數", "下轄鄉數", "人口消長", "地理分區"]
rename_list = [{"區名": "Chinese Name"}, {"面積 (km²)": "Area"}, {"人口數": "Population"}, {"人口密度 (人/km²)": "Density"}, {"郵遞區號": "Postal Code"}]

df = getDataFromWeb.get_taichung_info()
df = dataProcess.drop_rename_dataframe(df, row_drop_list, column_drop_list, rename_list)
df
```

Out[61]:

	Chinese Name	Area	Population	Density	Postal Code
0	中區	0.8803	18173	20644	400.0
1	東區	9.2855	75769	8160	401.0
2	南區	6.8101	126585	18588	402.0
3	西區	5.7042	114618	20094	403.0
4	北區	6.9376	146676	21142	404.0
5	北屯區	62.7034	286208	4564	406.0
6	西屯區	39.8467	230978	5797	407.0
7	南屯區	31.2578	175439	5613	408.0
8	太平區	120.7473	194835	1614	411.0
9	大里區	28.8759	212957	7375	412.0
10	霧峰區	98.0779	65144	664	413.0
11	烏日區	43.4032	76249	1757	414.0
12	豐原區	41.1845	166199	4035	420.0
13	后里區	58.9439	54374	922	421.0
14	石岡區	18.2105	14659	805	422.0
15	東勢區	117.4065	49250	419	423.0
16	和平區	1037.8192	10815	10	424.0
17	新社區	68.8874	24108	350	426.0
18	潭子區	25.8497	109274	4227	427.0
19	大雅區	32.4109	95703	2953	428.0
20	神岡區	35.0445	65550	1870	429.0
21	大肚區	37.0024	56759	1534	432.0
22	沙鹿區	40.4604	95135	2351	433.0
23	龍井區	38.0377	77998	2051	434.0
24	梧棲區	16.6049	59199	3565	435.0
25	清水區	64.1709	87863	1369	436.0
26	大甲區	58.5192	76116	1301	437.0
27	外埔區	42.4098	31989	754	438.0
28	大安區	27.4045	18925	691	439.0

2.3 English - Chinese glossary of districts in Taiwan:

Data source: <http://gn.moi.gov.tw/geonames/Translation/Translation.aspx>

Description: Government help to translate the city or districts name between English and Chinese

```
In [12]: taichung_english_name_df = csvProcess.load_from_csv("Taichung_English_Name.csv")
print(taichung_english_name_df.shape)
taichung_english_name_df.head(30)
```

(29, 2)

Out[12]:

	Chinese Name	Name
0	南屯區	Nantun District
1	西屯區	Xitun District
2	北屯區	Beitun District
3	西區	West District
4	北區	North District
5	霧峰區	Wufeng District
6	豐原區	Fengyuan District
7	龍井區	Longjing District
8	潭子區	Tanzi District
9	新社區	Xinshe District
10	清水區	Qingshui District
11	梧棲區	Wuqi District
12	神岡區	Shengang District
13	烏日區	Wuri District
14	東勢區	Dongshi District
15	和平區	Heping District
16	和平區	Heping District
17	沙鹿區	Shalu District
18	后里區	Houli District
19	石岡區	Shigang District
20	外埔區	Waipu District
21	太平區	Taiping District
22	大雅區	Daya District
23	大里區	Dali District
24	大肚區	Dadu District
25	大安區	Da'an District
26	大甲區	Dajia District
27	中區	Central District
28	東區	East District
29	南區	South District

2.4 Latitude and longitude in Taiwan:

Data source:

<https://www.astrocode.net/%E5%8F%B0%E7%81%A3%E5%90%84%E7%B8%A3%E5%B8%82%E5%9C%B0%E5%8D%80%E7%B6%93%E7%B7%AF%E5%BA%A6/>

Description: This is a XML file and we can transfer to JSON file. The file included all districts of latitude and longitude in Taiwan.

```
In [13]: taiwan_geo_dict = jsonProcess.load_from_json("Taiwan_Geospatial_Coordinates.json")
taiwan_geo_dict
```

Out[13]:

```
{'dataroot': {'@_noNamespaceSchemaLocation': '1050429_行政區經緯度(toPost).xsd',
 '@_generated': '2016-08-12T12:06:09',
 '_x0031_050429_行政區經緯度_x0028_toPost_x0029_': [{'行政區名': '臺北市中正區',
 '_x0033_ 碼郵遞區號': '100',
 '中心點經度': '121.5198839',
 '中心點緯度': '25.03240487',
 'TGOS_URL': 'http://tgos.nat.gov.tw/tgos/Web/MetaData/TGOS_MetaData_View.aspx?MID=9C715A5CD330360D355AE105F908B29E&SHOW_BACK_BUTTON=false'},
 {'行政區名': '臺北市大同區',
 '_x0033_ 碼郵遞區號': '103',
 '中心點經度': '121.5130417',
 '中心點緯度': '25.06342433',
 'TGOS_URL': 'http://tgos.nat.gov.tw/tgos/Web/MetaData/TGOS_MetaData_View.aspx?MID=9C715A5CD330360D355AE105F908B29E&SHOW_BACK_BUTTON=false'},
 {'行政區名': '臺北市中山區',
 '_x0033_ 碼郵遞區號': '104',
 '中心點經度': '121.5381597',
 '中心點緯度': '25.06969917',
 'TGOS_URL': 'http://tgos.nat.gov.tw/tgos/Web/MetaData/TGOS_MetaData_View.aspx?MID=9C715A5CD330360D355AE105F908B29E&SHOW_BACK_BUTTON=false'}
```

2.5 Venues in each neighborhood of Taichung City:

Data source: Foursquare APIs

Description: We will get all the venues in each neighborhood by using this API and filter these venues to get restaurants and other data.