MSPA PREDICT 420

Graded Exercise 4: Customers of a Hotel Speak

Introduction

This document presents the results of the forth graded exercise for the Masters of Science in Predictive Analytics course: PREDICT 420. This assessment required the student to work with json formatted TripAdvisor customer review data in order to help understand the feedback from aggregated reviews.

Assessment

1. Loading the Data

In [1]:
```
import json

with open("data/100506.json") as input_file:
    jsondat = json.load(input_file)
```

2. Extract Ratings Data

Extract ratings data into a list of dictionaries and convert to a dataframe.

In [2]:
```
import pandas

#Source: http://stackoverflow.com/questions/38987/how-can-i-merge-two-python-dictionaries-in-a-single-expression
def merge_dicts(*dict_args):
    result = {}
    for dictionary in dict_args:
        result.update(dictionary)
    return result

ratingdictlist = [] # Create ratings dictionary list.
ratingfield = ["Service", # Exhaustive list of rating fields to extract. Note that fields not consistent between records.
               "Sleep Quality",
               "Check in / front desk",
               "Rooms",
               "Cleanliness",
               "Location",
               "Business service (e.g., internet access)",
               "Value",
               "Overall"]

for i in jsondat["Reviews"]: # Loop through each record in 'Reviews'.
    expdict = {"Author" : i["Author"], # Extract 'Author' and 'Date' from first level of record.
               "Date" : i["Date"]}

    for r in ratingfield:
        try: # Try to extract exhaustive list of ratings from second level (of 'Ratings').
            ratingdict = {r : i["Ratings"][r]}
        except KeyError: # NaN if ratings field does not exist for that record.
            ratingdict = {r : "NaN"}

        expdict = merge_dicts(expdict, ratingdict) # Merge extracted ratings to current record dictonary.

    ratingdictlist.append(expdict) # Append current record dictonary to list of dictionaries.

df_ratings = pandas.DataFrame(ratingdictlist) # Convert list of dictionaries to pandas dataframe.
```

Rename dataframe fields.

In [3]:
```
df_ratings.columns = ["authorName",
                      "busservicesRate",
                      "fontdeskRate",
                      "cleanlinessRate",
                      "reviewDate",
                      "locationRate",
                      "overallRate",
                      "roomRate",
                      "serviceRate",
                      "sleepRate",
                      "valueRate"]
```

Convert numeric fields to correct dtype.

In [4]:
```
numfields = ["busservicesRate", # Identify numeric fields.
             "fontdeskRate",
             "cleanlinessRate",
             "locationRate",
             "overallRate",
             "roomRate",
             "serviceRate",
             "sleepRate",
             "valueRate"]

df_ratings[numfields] = df_ratings[numfields].apply(pandas.to_numeric, errors = "coerce") # Conv
ert numeric fields to numeric dtype.
```

Confirm dtypes for created dataframe.

In [5]:
```
print(df_ratings.dtypes)
```

```
authorName          object
busservicesRate    float64
fontdeskRate       float64
cleanlinessRate    float64
reviewDate          object
locationRate       float64
overallRate        float64
roomRate           float64
serviceRate        float64
sleepRate          float64
valueRate          float64
dtype: object
```

Set dataframe index to 'authorName'.

In [6]:
```
df_ratings = df_ratings.set_index("authorName")
df_ratings.index.name = None
```

Print first five records of created dataframe.

In [7]:
```
df_ratings.head(5)
```

Out[7]:

	busservicesRate	fontdeskRate	cleanlinessRate	reviewDate	locationRate	overallRate	roomRate	serviceRat
luvsroadtrips	NaN	NaN	1	January 3, 2012	5	1	1	1
estelle e	NaN	NaN	4	December 29, 2011	5	4	3	4
RobertEddy	NaN	NaN	2	December 20, 2011	1	1	1	1
James R	NaN	NaN	1	October 30, 2011	1	1	1	1
Shobha49	NaN	NaN	NaN	September 14, 2011	5	1	1	1

3. Calculate Ratings Data Statistics

Calculate the minimum value for each rating.

In [8]:
```
pandas.DataFrame.min(df_ratings[numfields])
```

Out[8]:
```
busservicesRate    1
fontdeskRate       1
cleanlinessRate    1
locationRate       1
overallRate        1
roomRate           1
serviceRate        1
sleepRate          1
valueRate          1
dtype: float64
```

Calculate the maximum value for each rating.

In [9]:
```
pandas.DataFrame.max(df_ratings[numfields])
```

Out[9]:
```
busservicesRate    1
fontdeskRate       5
cleanlinessRate    5
locationRate       5
overallRate        4
roomRate           5
serviceRate        5
sleepRate          5
valueRate          5
dtype: float64
```

Calculate the mean value for each rating.

In [10]:
```
pandas.DataFrame.mean(df_ratings[numfields])
```

Out[10]:
```
busservicesRate    1.000000
fontdeskRate       3.000000
cleanlinessRate    2.000000
locationRate       4.000000
overallRate        1.666667
roomRate           1.545455
serviceRate        2.300000
sleepRate          2.176471
valueRate          2.000000
dtype: float64
```

4. Extract Comments Data

In [11]:
```
import pandas

def merge_dicts(*dict_args):
    result = {}
    for dictionary in dict_args:
        result.update(dictionary)
    return result

contentdictlist = [] # Create comments dictionary list.

for i in jsondat["Reviews"]: # Loop through each record in 'Reviews'.
    expdict = {"Author" : i["Author"], # Extract 'Author', 'Date' and 'Content' from first leve
l of record.
               "Date" : i["Date"],
               "Content" : i["Content"]}

    contentdictlist.append(expdict) # Append current record dictonary to list of dictionaries.

df_content = pandas.DataFrame(contentdictlist) # Convert list of dictionaries to pandas datafra
me.
```

Rename dataframe fields.

In [12]:
```
df_content.columns = ["authorName",
                      "commentString",
                      "reviewDate"]
```

Confirm dtypes for created dataframe.

In [13]:
```
print(df_content.dtypes)
```

```
authorName       object
commentString    object
reviewDate       object
dtype: object
```

Set dataframe index to 'authorName'.

In [14]:
```
df_content = df_content.set_index("authorName")
df_content.index.name = None
```

Print first five records of created dataframe.

In [15]:
```
df_content.head(5)
```

Out[15]:

	commentString	reviewDate
luvsroadtrips	This place is not even suitable for the homele...	January 3, 2012
estelle e	We stayed in dow ntow n hotel Seattle for tw o ni...	December 29, 2011
RobertEddy	i made reservations and w hen i show ed up, i qu...	December 20, 2011
James R	This hotel is so bad it's a joke. I could bare...	October 30, 2011
Shobha49	My husband and I stayed at this hotel from 16t...	September 14, 2011

5. Pickle Final Dataframes

In [16]:
```
import pickle

df_ratings.to_pickle("data/ratings.p")
df_content.to_pickle("data/content.p")
```

6. Extract Hotel Information

Create hotel information extraction function.

In [17]:
```
import json
from html.parser import HTMLParser
import pandas

#Source: http://stackoverflow.com/questions/753052/strip-html-from-strings-in-python
class MLStripper(HTMLParser):
    def __init__(self):
        self.reset()
        self.strict = False
        self.convert_charrefs= True
        self.fed = []
    def handle_data(self, d):
        self.fed.append(d)
    def get_data(self):
        return ''.join(self.fed)

def strip_tags(html):
    s = MLStripper()
    s.feed(html)
    return s.get_data()

def returninfo(data):
    infodict = {} # Create info dictionary.

    with open(data) as input_file:
        jsondat = json.load(input_file)

    for i in jsondat["HotelInfo"].keys(): # Loop through keys within 'HotelInfo'.
        j = strip_tags(jsondat["HotelInfo"][i]) # Strip HTML formatting for each keys record.
        infodict[i] = j

    df_content = pandas.DataFrame(infodict, index = [0]) # Convert dictonary to dataframe.

    return df_content
```

Apply extraction function to provided json files.

In [18]:
```
data = ["data/100506.json",
        "data/677703.json",
        "data/1217974.json"]
```

In [19]:
```
returninfo(data[0])
```

Out[19]:

	Address	HotelID	HotelURL	ImgURL	Name	Price
0	315 Seneca St., Seattle, WA 98101	100506	/Show UserReview s-g60878-d100506-Review s-Hotel_...	http://media-cdn.tripadvisor.com/media/Provide...	Hotel Seattle	96—118*

In [20]:
```
returninfo(data[1])
```

Out[20]:

	HotelID	HotelURL	Price
0	677703	http://w w w .tripadvisor.com/Show UserReview s-g15...	Unkonw n

In [21]:
```
returninfo(data[2])
```

Out[21]:

	HotelID	HotelURL	Price
0	1217974	http://w w w .tripadvisor.com/Show UserReview s-g60...	Unkonw n