Muley_Tushar_Week5_Assignment_5_2

July 10, 2021

Name: Muley, Tushar

Assignment: Week 5 Assignment 5.2 Hotel Recommendation

Date: July 11, 2021

Description: Online travel agencies are scrambling to meet the artificial intelligence driven personalization standard set by companies like Amazon and Netflix. In addition, the world of online travel has become a highly competitive space where brands try to capture our attention (and wallet) with recommending, comparing, matching, and sharing. For this assignment, we would like to create the optimal hotel recommendations for Expedia's users that are searching for a hotel to book.

To get started, I would suggest exploring the file train.csv, which contains the logs of user behavior. There is another file named destinations.csv, which contains information related to hotel reviews made by users.

Build at least two prediction models from the training set, and report the accuracies on the test set.

[41]:
```
#Load common libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import csv
import sklearn
import random
import operator
import datetime

from sklearn.decomposition import PCA
from sklearn import preprocessing
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.pipeline import make_pipeline
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn import svm
from sklearn.model_selection import train_test_split
```

[2]:
```
#Increase sire of view window
pd.set_option('display.max_columns', None)
```

[5]:
```
# run once to get data than copy to pc
#train = pd.read_csv('train.csv', iterator = True).get_chunk(10000000).dropna()
#train = pd.read_csv('train.csv', sep=',').dropna()
#dest = pd.read_csv('destinations.csv')
#train = train.sample(frac=0.01, random_state=99)
#train.shape
```

[5]:
```
(241179, 24)
```

[6]:
```
# Write file down to csv
train.to_csv('train_lite.csv', index=False)
```

[15]:
```
# csv file name
file = 'destinations.csv'

# initializing the titles and rows list
fields = []
rows = []

# reading csv file
with open(file, 'r') as csvfile:
    # creating a csv reader object
    csvreader = csv.reader(csvfile)

    # extracting field names through first row
    fields = next(csvreader)

    # extracting each data row one by one
    for row in csvreader:
        rows.append(row)

    # get total number of rows
    print("Total no. of rows: %d"%(csvreader.line_num))
```

```
Total no. of rows: 62107
```

[3]:
```
# bring back files for analysis
dest = pd.read_csv('destinations.csv')
train = pd.read_csv('train_lite.csv')
```

[4]:
```
print('The dimension of the destnation table : ', dest.shape)

print('The dimension of the train table : ', train.shape)
```

```
The dimension of the destnation table :  (62106, 150)
The dimension of the train table :  (241179, 24)
```

[5]:
```
print("Summarized Data")
print(train.describe(include=['O']))
```

```
Summarized Data
                  date_time     srch_ci     srch_co
count                241179      241179      241179
unique               240445        1085        1094
top     2014-11-02 09:46:33  2014-12-26  2014-12-28
freq                      3        1566        1364
```

[49]:
```
# view data in destination
print(destinations.head(5))
```

```
   srch_destination_id        d1        d2        d3        d4        d5  \
0                    0 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657
1                    1 -2.181690 -2.181690 -2.181690 -2.082564 -2.181690
2                    2 -2.183490 -2.224164 -2.224164 -2.189562 -2.105819
3                    3 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409
4                    4 -2.189562 -2.187783 -2.194008 -2.171153 -2.152303

         d6        d7        d8        d9       d10       d11       d12  \
0 -1.897627 -2.198657 -2.198657 -1.897627 -2.198657 -2.198657 -2.198657
1 -2.165028 -2.181690 -2.181690 -2.031597 -2.181690 -2.181690 -2.181690
2 -2.075407 -2.224164 -2.118483 -2.140393 -2.224164 -2.209855 -2.224164
3 -2.115485 -2.177409 -2.177409 -2.177409 -2.177409 -2.161081 -2.177409
4 -2.056618 -2.194008 -2.194008 -2.145911 -2.194008 -2.089094 -2.194008

        d13       d14       d15       d16       d17       d18       d19  \
0 -2.198657 -2.198657 -2.198657 -1.897627 -2.198657 -2.198657 -2.198657
1 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690
2 -2.110723 -2.186008 -2.224164 -2.124474 -2.224164 -2.156467 -2.224164
3 -2.177409 -2.177409 -2.177409 -2.130158 -2.177409 -2.177409 -2.177409
4 -2.155205 -2.070995 -2.194008 -2.074964 -2.185526 -2.194008 -2.189562

        d20       d21       d22       d23       d24       d25       d26  \
0 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657
1 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690
2 -2.224164 -2.224164 -2.224164 -2.224164 -2.224164 -2.224164 -2.224164
3 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409
4 -2.191779 -2.185032 -2.150215 -2.194008 -2.189233 -2.194008 -2.191631

        d27       d28       d29       d30       d31       d32       d33  \
0 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657
1 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690
2 -2.224164 -2.224164 -2.153316 -2.186008 -2.224164 -2.224164 -2.224164
3 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409
4 -2.146025 -2.194008 -2.130263 -2.177813 -2.194008 -2.159651 -2.194008
```

```
        d34       d35       d36       d37       d38       d39       d40  \
0 -2.198657 -2.198657 -2.198657 -1.897627 -2.198657 -2.198657 -2.198657
1 -2.181690 -2.181690 -2.181690 -2.043789 -2.181690 -2.181690 -2.181690
2 -2.128237 -2.158309 -2.059716 -1.910270 -2.224164 -2.123050 -2.224164
3 -2.177409 -2.177409 -2.177409 -2.101291 -2.177409 -2.177409 -2.177409
4 -2.170700 -2.194008 -2.194008 -1.946523 -2.191779 -2.194008 -2.172218

        d41       d42       d43       d44       d45       d46       d47  \
0 -2.198657 -2.198657 -1.897627 -2.198657 -2.198657 -2.198657 -2.198657
1 -2.165028 -2.181690 -2.133508 -2.181690 -2.181690 -2.181690 -2.181690
2 -2.154047 -2.197327 -1.945461 -2.224164 -2.224164 -2.197327 -2.166886
3 -2.177409 -2.177409 -2.145344 -2.177409 -2.177409 -2.177409 -2.145344
4 -2.188901 -2.194008 -2.109317 -2.181161 -2.175595 -2.194008 -2.180677

        d48       d49       d50       d51       d52       d53       d54  \
0 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657
1 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690 -2.133508 -2.181690
2 -2.192009 -1.903782 -2.224164 -2.189562 -2.224164 -2.083104 -2.224164
3 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409
4 -2.194008 -2.092435 -2.194008 -2.172544 -2.194008 -2.145418 -2.194008

        d55       d56       d57       d58       d59       d60       d61  \
0 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657
1 -2.181690 -2.181690 -2.181690 -2.082564 -2.181690 -2.181690 -2.181690
2 -2.070096 -2.157745 -2.196379 -2.069813 -2.224164 -2.224164 -2.072953
3 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409
4 -2.115699 -2.120789 -2.188037 -2.149211 -2.194008 -2.194008 -2.191779

        d62       d63       d64       d65       d66       d67       d68  \
0 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657
1 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690
2 -2.224164 -2.224164 -2.224164 -2.224164 -2.140393 -2.224164 -2.224164
3 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409
4 -2.194008 -2.185032 -2.194008 -2.194008 -2.169871 -2.194008 -2.194008

        d69       d70       d71       d72       d73       d74       d75  \
0 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657
1 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690
2 -2.096550 -2.224164 -2.187529 -2.224164 -2.224164 -2.098667 -2.016373
3 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409
4 -2.167989 -2.194008 -2.158422 -2.194008 -2.191779 -2.174543 -2.180712

        d76       d77       d78       d79       d80       d81       d82  \
0 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657
1 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690
2 -2.224164 -2.126123 -2.189562 -2.008317 -2.085827 -2.224164 -2.224164
3 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409
4 -2.177944 -2.188901 -2.165486 -2.191779 -2.145961 -2.194008 -2.187356
```

```
        d83       d84       d85       d86       d87       d88       d89  \
0 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657
1 -2.181690 -2.165028 -2.165028 -2.181690 -2.181690 -2.181690 -2.181690
2 -2.224164 -2.218361 -2.219342 -2.224164 -2.224164 -2.160492 -2.224164
3 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409
4 -2.169871 -2.191715 -2.191110 -2.194008 -2.194008 -2.176134 -2.194008

        d90       d91       d92       d93       d94       d95       d96  \
0 -2.198657 -2.198657 -2.198657 -1.897627 -2.198657 -2.198657 -2.198657
1 -2.181690 -2.082564 -2.181690 -2.133508 -2.181690 -2.181690 -2.181690
2 -2.224164 -2.076796 -2.224164 -1.975081 -2.166149 -2.224164 -2.224164
3 -2.177409 -2.177409 -2.177409 -2.101291 -2.177409 -2.177409 -2.177409
4 -2.194008 -2.194008 -2.194008 -1.964288 -2.176791 -2.191779 -2.191779

        d97       d98       d99      d100      d101      d102      d103  \
0 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657
1 -2.181690 -2.082564 -2.181690 -2.181690 -2.181690 -2.181690 -2.165028
2 -2.224164 -2.224164 -2.224164 -2.163748 -2.224164 -2.072359 -2.100277
3 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409
4 -2.189562 -2.194008 -2.194008 -2.194008 -2.165889 -2.194008 -2.145988

       d104      d105      d106      d107      d108      d109      d110  \
0 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657
1 -2.181690 -2.181690 -2.181690 -2.181690 -2.148982 -2.181690 -2.181690
2 -2.224164 -2.203533 -2.224164 -2.224164 -2.094364 -2.224164 -2.140393
3 -2.177409 -2.177409 -2.177409 -2.177409 -2.161081 -2.177409 -2.177409
4 -2.194008 -2.194008 -2.191779 -2.194008 -2.097420 -2.194008 -2.180027

       d111      d112      d113      d114      d115      d116      d117  \
0 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657
1 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690 -2.181690
2 -2.224164 -2.164722 -2.224164 -2.154997 -2.197327 -2.224164 -2.197327
3 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409
4 -2.194008 -2.194008 -2.194008 -2.194008 -2.191779 -2.194008 -2.194008

       d118      d119      d120      d121      d122      d123      d124  \
0 -2.198657 -2.198657 -2.198657 -1.897627 -2.198657 -2.198657 -2.198657
1 -2.181690 -2.181690 -2.181690 -2.181690 -2.133508 -2.181690 -2.181690
2 -2.224164 -2.224164 -2.197327 -2.187005 -2.026263 -2.224164 -2.224164
3 -2.177409 -2.177409 -2.177409 -2.115485 -2.177409 -2.177409 -2.177409
4 -2.194008 -2.194008 -2.194008 -2.152740 -2.139178 -2.194008 -2.182305

       d125      d126      d127      d128      d129      d130      d131  \
0 -2.198657 -2.198657 -2.198657 -1.897627 -2.198657 -2.198657 -2.198657
1 -2.181690 -2.181690 -2.181690 -2.133508 -2.181690 -2.181690 -2.181690
2 -2.224164 -2.223818 -2.224164 -2.049280 -2.189562 -2.118483 -2.145558
3 -2.177409 -2.177409 -2.177409 -2.161081 -2.177409 -2.177409 -2.161081
```

```
4 -2.194008 -2.194008 -2.194008 -2.157163 -2.194008 -2.174454 -2.139365

       d132      d133      d134      d135      d136      d137      d138  \
0 -1.897627 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657
1 -2.148982 -2.181690 -2.181690 -2.181690 -2.181690 -2.148982 -2.181690
2 -2.117811 -2.224164 -2.180182 -2.224164 -2.224164 -2.214572 -2.186008
3 -2.130158 -2.177409 -2.177409 -2.177409 -2.177409 -2.115485 -2.177409
4 -2.099302 -2.194008 -2.194008 -2.194008 -2.194008 -2.151470 -2.194008

       d139      d140      d141      d142      d143      d144      d145  \
0 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657 -2.198657
1 -2.181690 -2.165028 -2.181690 -2.165028 -2.181690 -2.181690 -2.165028
2 -2.191569 -2.224164 -2.224164 -2.196379 -2.224164 -2.192009 -2.224164
3 -2.161081 -2.161081 -2.177409 -2.177409 -2.177409 -2.177409 -2.177409
4 -2.163242 -2.187356 -2.194008 -2.191779 -2.194008 -2.194008 -2.185161

       d146      d147      d148      d149
0 -2.198657 -2.198657 -2.198657 -2.198657
1 -2.181690 -2.181690 -2.181690 -2.181690
2 -2.224164 -2.224164 -2.224164 -2.057548
3 -2.177409 -2.177409 -2.177409 -2.177409
4 -2.194008 -2.194008 -2.194008 -2.188037
```

[8]:
```
# view data
print(train.head(5))
```

```
                   date_time  site_name  posa_continent  \
32352134  2014-05-22 11:40:07          2               3
29796021  2013-06-29 12:24:37          2               3
15185156  2014-10-30 13:58:32          2               3
3301948   2014-08-22 20:14:34          2               3
25429119  2014-03-25 18:47:43          2               3

          user_location_country  user_location_region  user_location_city  \
32352134                     66                   174               24103
29796021                     66                   311               25538
15185156                     66                   294               40046
3301948                      66                   332               55121
25429119                     66                   314               47869

          orig_destination_distance  user_id  is_mobile  is_package  …  \
32352134                  2323.5232   802499          0           1  …
29796021                  2288.6121    85229          0           0  …
15185156                   587.6970   755217          0           1  …
3301948                   2234.4394   160733          0           1  …
25429119                   839.0087  1078493          0           0  …

          srch_children_cnt srch_rm_cnt srch_destination_id  \
```

```
32352134                   0           1                1442
29796021                   1           1                8272
15185156                   0           1               11321
3301948                    0           1                1152
25429119                   0           1                8284

          srch_destination_type_id  is_booking  cnt  hotel_continent  \
32352134                         3           0    1                4
29796021                         1           0    1                2
15185156                         1           0    1                2
3301948                          1           1    1                4
25429119                         1           0    4                2

          hotel_country  hotel_market  hotel_cluster
32352134            125           177             44
29796021             50           659             59
15185156             50           642             22
3301948              47          1502             65
25429119             50           685              6

[5 rows x 24 columns]
```

[9]:
```
print('Describe Data')
print(train.describe())
```

```
Describe Data
           site_name  posa_continent  user_location_country  \
count  241179.000000   241179.000000          241179.000000
mean        6.219650        2.902972              87.833837
std         9.016623        0.537927              54.548928
min         2.000000        0.000000               0.000000
25%         2.000000        3.000000              66.000000
50%         2.000000        3.000000              66.000000
75%         2.000000        3.000000              66.000000
max        53.000000        4.000000             215.000000

       user_location_region  user_location_city  orig_destination_distance  \
count         241179.000000       241179.000000              241179.000000
mean             311.466504        27987.797437                1970.282004
std              143.925625        16527.400909                2232.031046
min              135.000000            0.000000                   0.005600
25%              174.000000        14542.000000                 313.583300
50%              321.000000        27655.000000                1140.493100
75%              363.000000        42899.000000                2551.720250
max             1021.000000        56507.000000               11760.169700

            user_id      is_mobile     is_package        channel  …  \
count  2.411790e+05  241179.000000  241179.000000  241179.000000  …
```

```
mean   5.982821e+05       0.135659       0.242318       6.023522  …
std    3.424769e+05       0.342426       0.428487       3.722500  …
min    1.500000e+01       0.000000       0.000000       0.000000  …
25%    3.049250e+05       0.000000       0.000000       2.000000  …
50%    5.968820e+05       0.000000       0.000000       9.000000  …
75%    8.934920e+05       0.000000       0.000000       9.000000  …
max    1.198776e+06       1.000000       1.000000      10.000000  …

       srch_children_cnt    srch_rm_cnt  srch_destination_id  \
count      241179.000000  241179.000000        241179.000000
mean            0.350101       1.104387         14403.073153
std             0.756646       0.438016         10831.844497
min             0.000000       0.000000             4.000000
25%             0.000000       1.000000          8267.000000
50%             0.000000       1.000000         11319.000000
75%             0.000000       1.000000         17859.000000
max             9.000000       8.000000         65102.000000

       srch_destination_type_id     is_booking            cnt  \
count             241179.000000  241179.000000  241179.000000
mean                   2.633600       0.083059       1.470957
std                    2.168411       0.275971       1.186757
min                    0.000000       0.000000       1.000000
25%                    1.000000       0.000000       1.000000
50%                    1.000000       0.000000       1.000000
75%                    5.000000       0.000000       2.000000
max                    9.000000       1.000000      45.000000

       hotel_continent  hotel_country   hotel_market  hotel_cluster
count    241179.000000  241179.000000  241179.000000  241179.000000
mean          2.894481      74.127320     620.723060      49.928609
std           1.498428      53.707363     475.778566      29.135992
min           0.000000       0.000000       0.000000       0.000000
25%           2.000000      50.000000     246.000000      25.000000
50%           2.000000      50.000000     628.000000      49.000000
75%           4.000000      77.000000     701.000000      73.000000
max           6.000000     212.000000    2117.000000      99.000000

[8 rows x 21 columns]
```

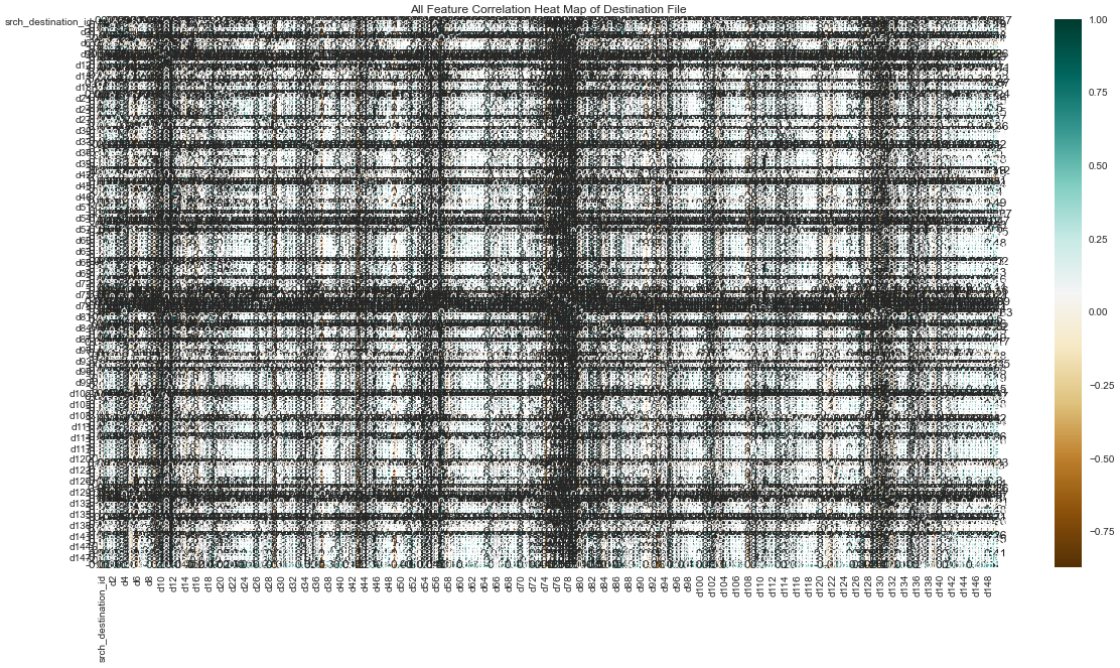
[11]:
```
print('Describe Data')
print(dest.describe())
```

```
Describe Data
       srch_destination_id            d1            d2            d3  \
count         62106.000000  62106.000000  62106.000000  62106.000000
mean          32359.463884     -2.193903     -2.202854     -2.207391
std           18711.765765      0.038576      0.041065      0.040092
```

```
min              0.000000     -2.597617     -2.671613     -2.671613
25%          16320.250000     -2.209336     -2.220192     -2.223679
50%          32277.500000     -2.185969     -2.193247     -2.195208
75%          48467.750000     -2.175309     -2.179261     -2.179990
max          64993.000000     -1.596004     -1.341325     -1.917826

                d4            d5            d6            d7            d8  \
count  62106.000000  62106.000000  62106.000000  62106.000000  62106.000000
mean      -2.194040     -2.161497     -2.045110     -2.202433     -2.203207
std        0.041406      0.066197      0.135803      0.038886      0.036578
min       -2.671613     -2.671613     -2.344165     -2.671613     -2.671613
25%       -2.211437     -2.192158     -2.154267     -2.218745     -2.219086
50%       -2.188057     -2.178255     -2.088598     -2.192513     -2.192915
75%       -2.176763     -2.153317     -1.975940     -2.178976     -2.179164
max       -1.814585     -1.607558     -1.209058     -1.474441     -1.879678

                d9  …          d140          d141          d142  \
count  62106.000000  …  62106.000000  62106.000000  62106.000000
mean      -2.107808  …     -2.204092     -2.196919     -2.203262
std        0.197904  …      0.037164      0.059914      0.046520
min       -2.495544  …     -2.620769     -2.671613     -2.671613
25%       -2.197704  …     -2.219630     -2.216147     -2.220347
50%       -2.177229  …     -2.192232     -2.190525     -2.192814
75%       -2.123598  …     -2.178093     -2.177777     -2.178996
max       -0.977219  …     -1.855317     -0.960356     -1.357408

               d143          d144          d145          d146          d147  \
count  62106.000000  62106.000000  62106.000000  62106.000000  62106.000000
mean      -2.205128     -2.201925     -2.203332     -2.202989     -2.208359
std        0.039465      0.041603      0.038239      0.051552      0.038035
min       -2.671613     -2.671613     -2.671613     -2.671613     -2.671613
25%       -2.221893     -2.218394     -2.220500     -2.221748     -2.224618
50%       -2.194127     -2.191564     -2.192200     -2.193352     -2.196185
75%       -2.179475     -2.178335     -2.177590     -2.178927     -2.180602
max       -1.775218     -1.790435     -1.799341     -1.335962     -1.816892

               d148          d149
count  62106.000000  62106.000000
mean      -2.208269     -2.199470
std        0.038569      0.042438
min       -2.671613     -2.671613
25%       -2.224290     -2.216428
50%       -2.195900     -2.190953
75%       -2.180380     -2.178088
max       -1.718778     -1.500309

[8 rows x 150 columns]
```

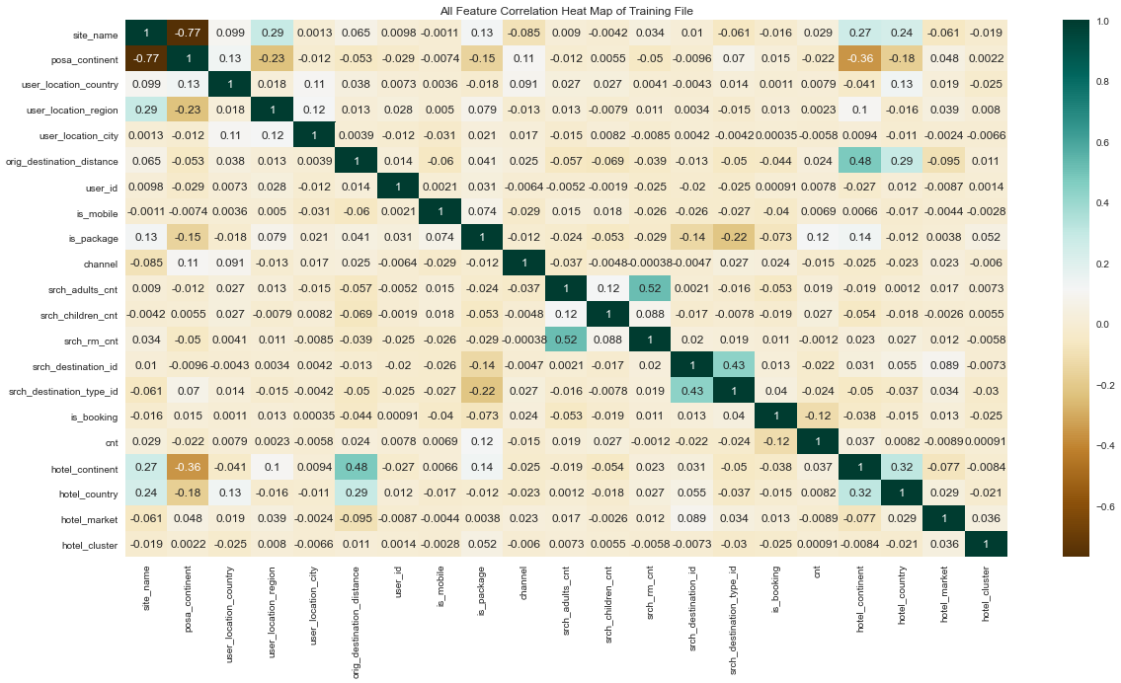
[53]:
```
#Run full correlation on the data set using heat map
plt.figure(figsize=(20,10))
c= dest.corr()
plt.title('All Feature Correlation Heat Map of Destination File')
plt.xlabel('Features')
plt.ylabel('Features')
sns.heatmap(c,cmap="BrBG",annot=True)
```

[53]:
```
<AxesSubplot:title={'center':'All Feature Correlation Heat Map of Destination
File'}>
```



[54]:
```
#Run full correlation on the data set using heat map
plt.figure(figsize=(20,10))
c= train.corr()
plt.title('All Feature Correlation Heat Map of Training File')
plt.xlabel('Features')
plt.ylabel('Features')
sns.heatmap(c,cmap="BrBG",annot=True)
```

[54]:
```
<AxesSubplot:title={'center':'All Feature Correlation Heat Map of Training
File'}>
```



[6]:
```
#Make a copy before dropping elements
train_copy1 = train.copy(deep=True)
```

[36]:
```
# use if needed
# copy the data in case need to go back
#train = train_copy1.copy(deep=True)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-36-4d453d1ae081> in <module>
      1 # use if needed
      2 # copy the data in case need to go back
----> 3 train = train_copy1.copy(deep=True)

NameError: name 'train_copy1' is not defined
```

[7]:
```
train.columns
```

[7]:
```
Index(['date_time', 'site_name', 'posa_continent', 'user_location_country',
       'user_location_region', 'user_location_city',
       'orig_destination_distance', 'user_id', 'is_mobile', 'is_package',
       'channel', 'srch_ci', 'srch_co', 'srch_adults_cnt', 'srch_children_cnt',
       'srch_rm_cnt', 'srch_destination_id', 'srch_destination_type_id',
       'is_booking', 'cnt', 'hotel_continent', 'hotel_country', 'hotel_market',
```

```
       'hotel_cluster'],
      dtype='object')
```

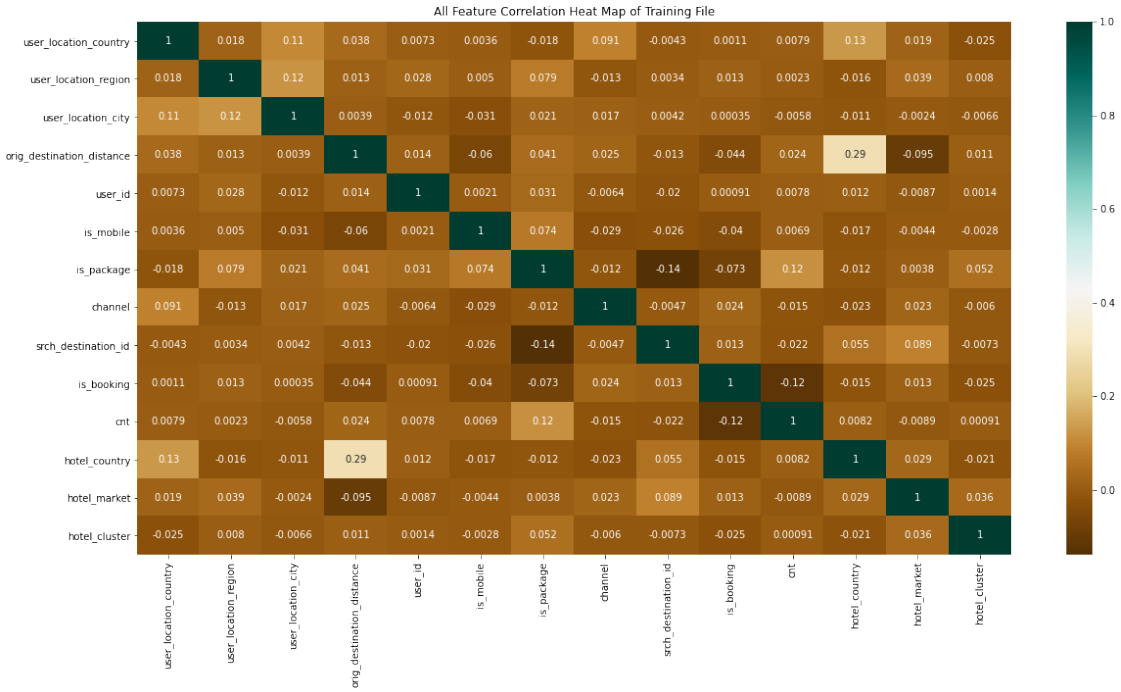
[6]:
```
# Elements to be dropped
train=train.drop(['posa_continent', 'site_name',␣
 ↪'srch_rm_cnt','srch_adults_cnt',
                        'srch_children_cnt','srch_destination_type_id',
                        'hotel_continent'], axis=1)
```

[7]:
```
df_train.columns
```

[7]:
```
Index(['date_time', 'user_location_country', 'user_location_region',
       'user_location_city', 'orig_destination_distance', 'user_id',
       'is_mobile', 'is_package', 'channel', 'srch_co', 'srch_destination_id',
       'is_booking', 'cnt', 'hotel_country', 'hotel_market', 'hotel_cluster'],
      dtype='object')
```

[8]:
```
# Run full correlation on the data set using heat map
plt.figure(figsize=(20,10))
c= train.corr()
plt.title('All Feature Correlation Heat Map of Training File')
plt.xlabel('Features')
plt.ylabel('Features')
sns.heatmap(c,cmap="BrBG",annot=True)
```

[8]:
```
<AxesSubplot:title={'center':'All Feature Correlation Heat Map of Training
File'}>
```



[59]:
```
train.head()
```

[59]:
```
             date_time  user_location_country  user_location_region  \
0  2014-08-11 07:46:59                     66                   348
1  2014-08-11 08:22:12                     66                   348
2  2014-08-11 08:24:33                     66                   348
3  2014-08-09 18:05:16                     66                   442
4  2014-08-09 18:08:18                     66                   442

   user_location_city  orig_destination_distance  is_mobile  is_package  \
0               48862                  2234.2641          0           1
1               48862                  2234.2641          0           1
2               48862                  2234.2641          0           0
3               35390                   913.1932          0           0
4               35390                   913.6259          0           0

   channel  srch_destination_id  is_booking  cnt  hotel_country  hotel_market  \
0        9                 8250           0    3             50           628
1        9                 8250           1    1             50           628
2        9                 8250           0    1             50           628
3        3                14984           0    1             50          1457
4        3                14984           0    1             50          1457

   hotel_cluster
0              1
1              1
2              1
3             80
4             21
```

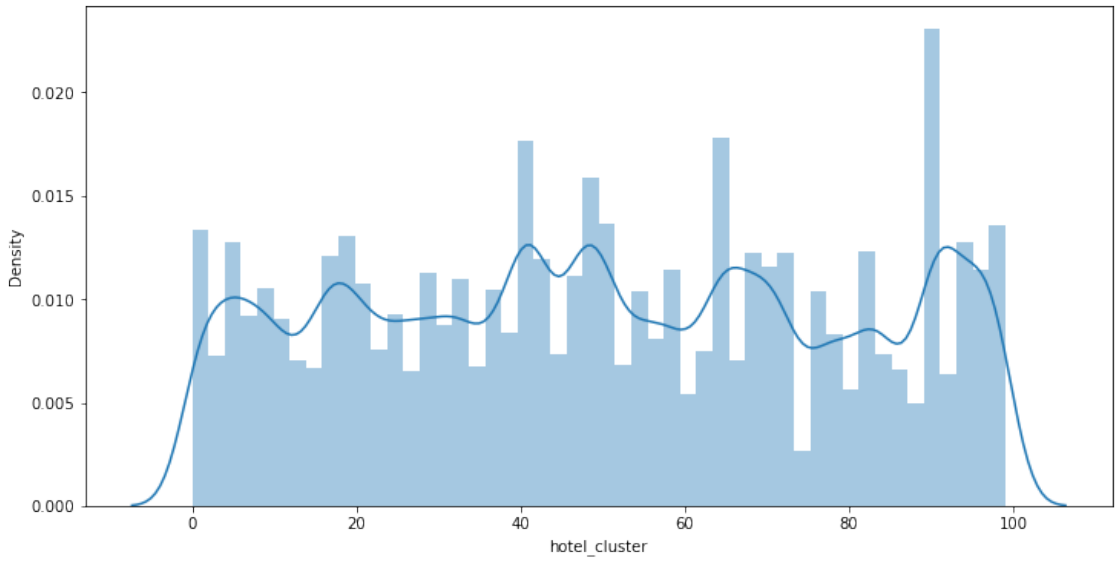
[60]:
```
train["hotel_cluster"].value_counts()
```

[60]:
```
91    2777
41    2176
48    2077
64    1984
65    1633
      …
35     439
53     359
27     332
88     301
74     125
Name: hotel_cluster, Length: 100, dtype: int64
```

[8]:
```
plt.figure(figsize=(12, 6))
sns.distplot(train['hotel_cluster'])
```

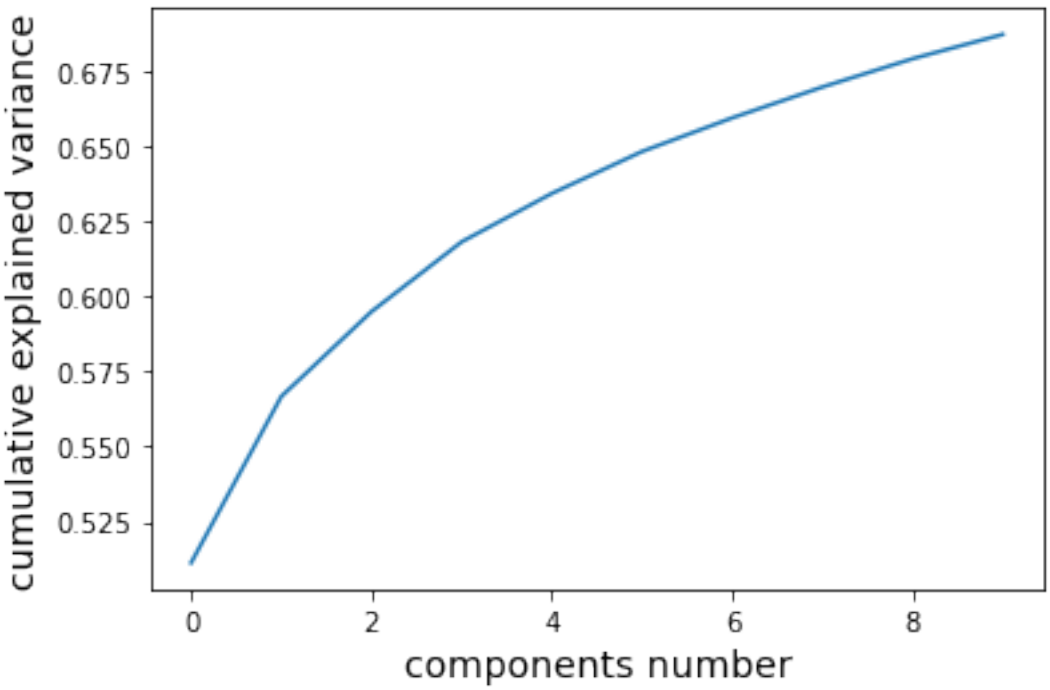
```
C:\Users\Tushar\anaconda3\lib\site-packages\seaborn\distributions.py:2557:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

[8]:
```
<AxesSubplot:xlabel='hotel_cluster', ylabel='Density'>
```



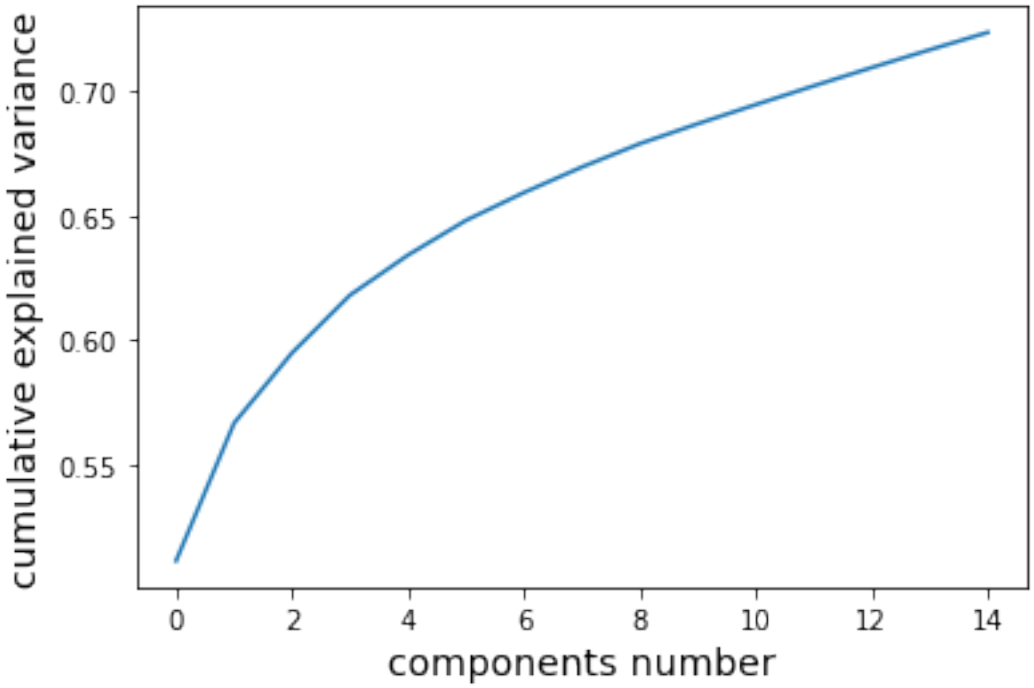
[23]:
```
# pca plot of features
pca = PCA(n_components=10)
scaler = preprocessing.StandardScaler()
destinations = scaler.fit_transform(dest)
pca.fit(destinations)
variance = pca.explained_variance_ratio_
print (variance)
cumulative_explained_variance = np.cumsum(variance)
plt.plot(cumulative_explained_variance)
plt.xlabel('components number', fontsize=14)
plt.ylabel('cumulative explained variance', fontsize=14)
plt.show()
```

```
[0.51135723 0.05529065 0.0282286  0.02320444 0.01609184 0.01393583
 0.01118764 0.01026042 0.00941039 0.00808163]
```



[54]:
```
# pca plot of features
pca = PCA(n_components=15)
scaler = preprocessing.StandardScaler()
destinations = scaler.fit_transform(dest)
pca.fit(destinations)
variance = pca.explained_variance_ratio_
print (variance)
cumulative_explained_variance = np.cumsum(variance)
plt.plot(cumulative_explained_variance)
plt.xlabel('components number', fontsize=14)
plt.ylabel('cumulative explained variance', fontsize=14)
plt.show()
```

```
[0.51135723 0.05529065 0.02822859 0.02320443 0.01609132 0.01393396
 0.01117312 0.01026071 0.00938215 0.00801961 0.00767812 0.00753141
 0.00734583 0.00715279 0.00696526]
```



[18]:
```
# check correlation of hotel_cluster

train.corr()["hotel_cluster"]
```

[18]:
```
site_name                  -0.027497
posa_continent              0.012180
user_location_country      -0.020239
user_location_region        0.006927
user_location_city          0.001241
orig_destination_distance   0.006084
user_id                     0.003891
is_mobile                   0.008788
is_package                  0.047598
channel                    -0.001386
srch_adults_cnt             0.012407
srch_children_cnt           0.014901
srch_rm_cnt                -0.005570
srch_destination_id        -0.016736
srch_destination_type_id   -0.036120
is_booking                 -0.022898
cnt                         0.000378
hotel_continent             0.000422
hotel_country              -0.023837
hotel_market                0.022149
```

```
hotel_cluster                1.000000
date_time_year              -0.000435
date_time_month             -0.002142
Name: hotel_cluster, dtype: float64
```

Note: No linear correlation of any kind.

1 Feature Engineering Models

This where I did my feature engineering with destination and training model.

If you are looking for my Original model it is below.

[10]:
```
# split out datetime field

from datetime import datetime
def get_year(x):
    if x is not None and type(x) is not float:
        try:
            return datetime.strptime(x, '%Y-%m-%d').year
        except ValueError:
            return datetime.strptime(x, '%Y-%m-%d %H:%M:%S').year
        else:
            return 2013
        pass

def get_month(x):
    if x is not None and type(x) is not float:
        try:
            return datetime.strptime(x, '%Y-%m-%d').month
        except:
            return datetime.strptime(x, '%Y-%m-%d %H:%M:%S').month
    else:
        return 1
    pass
```

[11]:
```
# merge the two df into one

def left_merge_dataset(left_dframe, right_dframe, merge_column):
    return pd.merge(left_dframe, right_dframe, on=merge_column, how='left')
```

[13]:
```
# new columns for year and month copy from original

train['date_time_year'] = pd.Series(train.date_time, index = train.index)
train['date_time_month'] = pd.Series(train.date_time, index = train.index)
```

[14]:
```
# view train
train.head()
```

[14]:

	date_time	site_name	posa_continent	user_location_country	user_location_region	user_location_city	orig_destination_distance	user_id	is_mobile	is_package	channel	srch_ci	srch_co	srch_adults_cnt	srch_children_cnt	srch_rm_cnt	srch_destination_id	srch_destination_type_id	is_booking	cnt	hotel_continent	hotel_country	hotel_market	hotel_cluster	date_time_year	date_time_month
0	2014-05-22 11:40:07	2	3	66	174	24103	2323.5232	802499	0	1	9	2014-07-24	2014-07-24	2	0	1	1442	3	0	1	4	125	177	44	2014-05-22 11:40:07	2014-05-22 11:40:07
1	2013-06-29 12:24:37	2	3	66	311	25538	2288.6121	85229	0	0	9	2013-07-05	2013-07-09	3	1	1	8272	1	0	1	2	50	659	59	2013-06-29 12:24:37	2013-06-29 12:24:37
2	2014-10-30 13:58:32	2	3	66	294	40046	587.6970	755217	0	1	9	2014-12-19	2014-12-22	2	0	1	11321	1	0	1	2	50	642	22	2014-10-30 13:58:32	2014-10-30 13:58:32
3	2014-08-22 20:14:34	2	3	66	332	55121	2234.4394	160733	0	1	9	2015-01-23	2015-01-30	2	0	1	1152	1	1	1	4	47	1502	65	2014-08-22 20:14:34	2014-08-22 20:14:34
4	2014-03-25 18:47:43	2	3	66	314	47869	839.0087	1078493	0	0	9	2014-04-17	2014-04-20	4	0	1	8284	1	0	4	2	50	685	6	2014-03-25 18:47:43	2014-03-25 18:47:43

[16]:
```
# conver year and month from datetime

from datetime import datetime # import date time to get year and month
```

```
train.date_time_year = train.date_time_year.apply(lambda x: get_year(x)) # for␣
↪year


train.date_time_month = train.date_time_month.apply(lambda x: get_month(x)) #␣
↪for month

del train['date_time'] #delete the original as it is not needed
```

[17]:
```
# view train date_time removed date_time_year replaced with year and␣
↪date_time_month replaced with month

train.head()
```

[17]:

	site_name	posa_continent	user_location_country	user_location_region	user_location_city	orig_destination_distance	user_id	is_mobile	is_package	channel	srch_ci	srch_co	srch_adults_cnt	srch_children_cnt	srch_rm_cnt	srch_destination_id	srch_destination_type_id	is_booking	cnt	hotel_continent	hotel_country
0	2	3	66	174	24103	2323.5232	802499	0	1	9	2014-07-24	2014-07-24	2	0	1	1442	3	0	1	4	125
1	2	3	66	311	25538	2288.6121	85229	0	0	9	2013-07-05	2013-07-09	3	1	1	8272	1	0	1	2	50
2	2	3	66	294	40046	587.6970	755217	0	1	9	2014-12-19	2014-12-22	2	0	1	11321	1	0	1	2	50
3	2	3	66	332	55121	2234.4394	160733	0	1	9	2015-01-23	2015-01-30	2	0	1	1152	1	1	1	4	47
4	2	3	66	314	47869	839.0087	1078493	0	0	9	2014-04-17	2014-04-20	4	0	1	8284	3	0	1	4	125

```
4                          1           0    4                 2             50

   hotel_market  hotel_cluster  date_time_year  date_time_month
0           177             44            2014                5
1           659             59            2013                6
2           642             22            2014               10
3          1502             65            2014                8
4           685              6            2014                3
```

[20]:
```
# get year and month for check in date

train['srch_ci_year'] = pd.Series(train.srch_ci, index=train.index)
train['srch_ci_month'] = pd.Series(train.srch_ci, index=train.index)

# convert year & months to int

train.srch_ci_year = train.srch_ci_year.apply(lambda x: get_year(x))
train.srch_ci_month = train.srch_ci_month.apply(lambda x: get_month(x))

# remove the srch_ci column
del train['srch_ci']
```

[21]:
```
# view train check in date_time removed date_time_year replaced with year and␣
 ↪date_time_month replaced with month

train.head()
```

[21]:
```
   site_name  posa_continent  user_location_country  user_location_region  \
0          2               3                     66                   174
1          2               3                     66                   311
2          2               3                     66                   294
3          2               3                     66                   332
4          2               3                     66                   314

   user_location_city  orig_destination_distance  user_id  is_mobile  \
0               24103                  2323.5232   802499          0
1               25538                  2288.6121    85229          0
2               40046                   587.6970   755217          0
3               55121                  2234.4394   160733          0
4               47869                   839.0087  1078493          0

   is_package  channel     srch_co  srch_adults_cnt  srch_children_cnt  \
0           1        9  2014-07-24                2                  0
1           0        9  2013-07-09                3                  1
2           1        9  2014-12-22                2                  0
3           1        9  2015-01-30                2                  0
4           0        9  2014-04-20                4                  0
```

```
   srch_rm_cnt  srch_destination_id  srch_destination_type_id  is_booking  \
0            1                 1442                         3           0
1            1                 8272                         1           0
2            1                11321                         1           0
3            1                 1152                         1           1
4            1                 8284                         1           0

   cnt  hotel_continent  hotel_country  hotel_market  hotel_cluster  \
0    1                4            125           177             44
1    1                2             50           659             59
2    1                2             50           642             22
3    1                4             47          1502             65
4    4                2             50           685              6

   date_time_year  date_time_month  srch_ci_year  srch_ci_month
0            2014                5          2014              7
1            2013                6          2013              7
2            2014               10          2014             12
3            2014                8          2015              1
4            2014                3          2014              4
```

[22]:
```
# get year and month for check out date

train['srch_co_year'] = pd.Series(train.srch_co, index=train.index)
train['srch_co_month'] = pd.Series(train.srch_co, index=train.index)

# convert year & months to int

train.srch_co_year = train.srch_co_year.apply(lambda x: get_year(x))
train.srch_co_month = train.srch_co_month.apply(lambda x: get_month(x))

# remove the srch_co column

del train['srch_co']
```

[23]:
```
# view train check out date_time removed date_time_year replaced with year and␣
↪date_time_month replaced with month

train.head()
```

[23]:
```
   site_name  posa_continent  user_location_country  user_location_region  \
0          2               3                     66                   174
1          2               3                     66                   311
2          2               3                     66                   294
3          2               3                     66                   332
4          2               3                     66                   314
```

```
   user_location_city  orig_destination_distance  user_id  is_mobile  \
0               24103                  2323.5232   802499          0
1               25538                  2288.6121    85229          0
2               40046                   587.6970   755217          0
3               55121                  2234.4394   160733          0
4               47869                   839.0087  1078493          0

   is_package  channel  srch_adults_cnt  srch_children_cnt  srch_rm_cnt  \
0           1        9                2                  0            1
1           0        9                3                  1            1
2           1        9                2                  0            1
3           1        9                2                  0            1
4           0        9                4                  0            1

   srch_destination_id  srch_destination_type_id  is_booking  cnt  \
0                 1442                         3           0    1
1                 8272                         1           0    1
2                11321                         1           0    1
3                 1152                         1           1    1
4                 8284                         1           0    4

   hotel_continent  hotel_country  hotel_market  hotel_cluster  \
0                4            125           177             44
1                2             50           659             59
2                2             50           642             22
3                4             47          1502             65
4                2             50           685              6

   date_time_year  date_time_month  srch_ci_year  srch_ci_month  srch_co_year  \
0            2014                5          2014              7          2014
1            2013                6          2013              7          2013
2            2014               10          2014             12          2014
3            2014                8          2015              1          2015
4            2014                3          2014              4          2014

   srch_co_month
0              7
1              7
2             12
3              1
4              4
```

Note: remove all the date fields and added new year and month fields for date_time, search checking and seach check out

[24]:
```
# creating a group to better determine correlation hote country and hotel␣
 ↪market will determine hotel cluster

pieces = [train.
 ↪groupby(['srch_destination_id','hotel_country','hotel_market','hotel_cluster'])
          ['is_booking'].agg(['sum','count'])]

agg = pd.concat(pieces).groupby(level=[0,1,2,3]).sum()
agg.dropna(inplace=True)
agg.head()
```

[24]:

				sum	count
srch_destination_id	hotel_country	hotel_market	hotel_cluster		
4	7	246	22	0	1
			29	0	1
			30	0	1
			32	1	2
			43	0	1

[25]:
```
# continue strategy of finding the hotel cluster

agg['sum_and_cnt'] = 0.85*agg['sum'] + 0.15*agg['count']
agg = agg.groupby(level=[0,1,2]).apply(lambda x: x.astype(float)/x.sum())
agg.reset_index(inplace=True)
agg.head()
```

[25]:

	srch_destination_id	hotel_country	hotel_market	hotel_cluster	sum	\
0	4	7	246	22	0.0	
1	4	7	246	29	0.0	
2	4	7	246	30	0.0	
3	4	7	246	32	1.0	
4	4	7	246	43	0.0	

	count	sum_and_cnt
0	0.125	0.073171
1	0.125	0.073171
2	0.125	0.073171
3	0.250	0.560976
4	0.125	0.073171

[26]:
```
# create the pivot table

agg_pivot = agg.
 ↪pivot_table(index=['srch_destination_id','hotel_country','hotel_market'],
                           columns='hotel_cluster', values='sum_and_cnt').
 ↪reset_index()
agg_pivot.head()
```

[26]:
```
hotel_cluster  srch_destination_id  hotel_country  hotel_market   0   1   2  \
0                                4              7           246 NaN NaN NaN
1                                8             50           416 NaN NaN NaN
2                               11             50           824 NaN NaN NaN
3                               14             27          1434 NaN NaN NaN
4                               16             50           419 NaN NaN NaN

hotel_cluster   3   4   5   6         7   8   9        10  11  12  13  14  15  \
0             NaN NaN NaN NaN       NaN NaN NaN       NaN NaN NaN NaN NaN NaN
1             NaN NaN NaN NaN  0.231092 NaN NaN  0.012605 NaN NaN NaN NaN NaN
2             NaN NaN NaN NaN       NaN NaN NaN       NaN NaN NaN NaN NaN NaN
3             NaN NaN NaN NaN       NaN NaN NaN       NaN NaN NaN NaN NaN NaN
4             NaN NaN NaN NaN  0.551724 NaN NaN       NaN NaN NaN NaN NaN NaN

hotel_cluster       16  17  18  19  20  21        22  23  24  25  26  27  28  \
0                  NaN NaN NaN NaN NaN NaN  0.073171 NaN NaN NaN NaN NaN NaN
1              0.02521 NaN NaN NaN NaN NaN       NaN NaN NaN NaN NaN NaN NaN
2                  NaN NaN NaN NaN NaN NaN       NaN NaN NaN NaN NaN NaN NaN
3                  NaN NaN NaN NaN NaN NaN       NaN NaN NaN NaN NaN NaN NaN
4                  NaN NaN NaN NaN NaN NaN       NaN NaN NaN NaN NaN NaN NaN

hotel_cluster        29        30  31        32  33  34  35  36  37  38  39  \
0              0.073171  0.073171 NaN  0.560976 NaN NaN NaN NaN NaN NaN NaN
1                   NaN       NaN NaN  0.012605 NaN NaN NaN NaN NaN NaN NaN
2                   NaN       NaN NaN       NaN NaN NaN NaN NaN NaN NaN NaN
3                   NaN       NaN NaN       NaN NaN NaN NaN NaN NaN NaN NaN
4                   NaN  0.051724 NaN       NaN NaN NaN NaN NaN NaN NaN NaN

hotel_cluster  40  41        42        43  44  45  46    47        48  49  50  \
0              NaN NaN       NaN  0.073171 NaN NaN NaN   NaN       NaN NaN NaN
1              NaN NaN  0.134454  0.121849 NaN NaN NaN   NaN  0.302521 NaN NaN
2              NaN NaN  0.500000       NaN NaN NaN NaN  0.25       NaN NaN NaN
3              NaN NaN       NaN       NaN NaN NaN NaN   NaN       NaN NaN NaN
4              NaN NaN       NaN       NaN NaN NaN NaN   NaN       NaN NaN NaN

hotel_cluster        51  52  53  54  55  56  57        58  59        60   61  \
0                   NaN NaN NaN NaN NaN NaN NaN  0.073171 NaN       NaN  NaN
1              0.096639 NaN NaN NaN NaN NaN NaN       NaN NaN  0.012605  NaN
2              0.250000 NaN NaN NaN NaN NaN NaN       NaN NaN       NaN  NaN
3                   NaN NaN NaN NaN NaN NaN NaN       NaN NaN       NaN  1.0
4                   NaN NaN NaN NaN NaN NaN NaN       NaN NaN       NaN  NaN

hotel_cluster  62  63  64  65  66  67  68  69  70  71  72  73  74  75  \
0              NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
1              NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
2              NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
3              NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
```

```
4              NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN

hotel_cluster        76  77  78  79  80  81        82  83  84  85  86  87  88  \
0                   NaN NaN NaN NaN NaN NaN  0.073171 NaN NaN NaN NaN NaN NaN
1              0.025210 NaN NaN NaN NaN NaN       NaN NaN NaN NaN NaN NaN NaN
2                   NaN NaN NaN NaN NaN NaN       NaN NaN NaN NaN NaN NaN NaN
3                   NaN NaN NaN NaN NaN NaN       NaN NaN NaN NaN NaN NaN NaN
4              0.051724 NaN NaN NaN NaN NaN       NaN NaN NaN NaN NaN NaN NaN

hotel_cluster  89  90        91  92  93  94  95  96  97  98  99
0              NaN NaN       NaN NaN NaN NaN NaN NaN NaN NaN NaN
1              NaN NaN  0.025210 NaN NaN NaN NaN NaN NaN NaN NaN
2              NaN NaN       NaN NaN NaN NaN NaN NaN NaN NaN NaN
3              NaN NaN       NaN NaN NaN NaN NaN NaN NaN NaN NaN
4              NaN NaN  0.344828 NaN NaN NaN NaN NaN NaN NaN NaN
```

[27]:
```
# merge data with destnation and train df

train = pd.merge(train, dest, how='left', on='srch_destination_id')
train = pd.merge(train, agg_pivot, how='left',␣
 ↪on=['srch_destination_id','hotel_country','hotel_market'])
train.fillna(0, inplace=True)
train.shape
```

[27]:
```
(241179, 276)
```

[28]:
```
# preview the new train df
train.head()
```

[28]:
```
   site_name  posa_continent  user_location_country  user_location_region  \
0          2               3                     66                   174
1          2               3                     66                   311
2          2               3                     66                   294
3          2               3                     66                   332
4          2               3                     66                   314

   user_location_city  orig_destination_distance  user_id  is_mobile  \
0               24103                  2323.5232   802499          0
1               25538                  2288.6121    85229          0
2               40046                   587.6970   755217          0
3               55121                  2234.4394   160733          0
4               47869                   839.0087  1078493          0

   is_package  channel  srch_adults_cnt  srch_children_cnt  srch_rm_cnt  \
0           1        9                2                  0            1
1           0        9                3                  1            1
2           1        9                2                  0            1
```

```
3           1        9                2                  0           1
4           0        9                4                  0           1

   srch_destination_id  srch_destination_type_id  is_booking  cnt  \
0                 1442                         3           0    1
1                 8272                         1           0    1
2                11321                         1           0    1
3                 1152                         1           1    1
4                 8284                         1           0    4

   hotel_continent  hotel_country  hotel_market  hotel_cluster  \
0                4            125           177             44
1                2             50           659             59
2                2             50           642             22
3                4             47          1502             65
4                2             50           685              6

   date_time_year  date_time_month  srch_ci_year  srch_ci_month  srch_co_year  \
0            2014                5          2014              7          2014
1            2013                6          2013              7          2013
2            2014               10          2014             12          2014
3            2014                8          2015              1          2015
4            2014                3          2014              4          2014

   srch_co_month        d1        d2        d3        d4        d5        d6  \
0              7 -2.107779 -2.225888 -2.206358 -2.264705 -2.092796 -2.008010
1              7 -2.243749 -2.257786 -2.250899 -2.219748 -2.128407 -1.875750
2             12 -2.062891 -2.087863 -2.231733 -2.231268 -2.097198 -2.141796
3              1 -2.106448 -2.274500 -2.274500 -2.272112 -2.240330 -1.855909
4              4 -2.220990 -2.088285 -2.298956 -2.312516 -2.034377 -1.783107

         d7        d8        d9       d10       d11       d12       d13  \
0 -2.264705 -2.264705 -1.667172 -2.197542 -2.249315 -2.264705 -2.030415
1 -2.165300 -2.223528 -2.244673 -2.217775 -2.126882 -2.243014 -2.245576
2 -2.260583 -2.224048 -2.160892 -2.190190 -2.212481 -2.261589 -2.253630
3 -2.274500 -2.274500 -1.448242 -2.248604 -2.209857 -2.274500 -1.984499
4 -2.214485 -2.310484 -1.316845 -2.220448 -2.111331 -2.314419 -1.974593

        d14       d15       d16       d17       d18       d19       d20  \
0 -2.231544 -2.264705 -2.227558 -2.264705 -2.264705 -2.264705 -2.264705
1 -2.059609 -2.258465 -1.893482 -2.225674 -2.247744 -2.257553 -2.141050
2 -2.025441 -2.261589 -2.121029 -2.244249 -2.246261 -2.261589 -2.261589
3 -2.170847 -2.274500 -2.256156 -2.178890 -2.269899 -2.272252 -2.266252
4 -2.200258 -2.313130 -2.126809 -2.255569 -2.247951 -2.247877 -2.312334

        d21       d22       d23       d24       d25       d26       d27  \
0 -2.264705 -2.264705 -2.264705 -2.251320 -2.264705 -2.264705 -2.264705
```

```
1 -2.225062 -2.258465 -2.258465 -2.246325 -2.258465 -2.229398 -2.258260
2 -2.136160 -2.261589 -2.260697 -2.253083 -2.261589 -2.257690 -2.137547
3 -2.274500 -2.274500 -2.273375 -2.272467 -2.274500 -2.258871 -2.274500
4 -2.312560 -2.314419 -2.314419 -2.309626 -2.314419 -2.296420 -2.176713

        d28       d29       d30       d31       d32       d33       d34  \
0 -2.243367 -2.180757 -2.264705 -2.175682 -2.261583 -2.264705 -2.049239
1 -2.237616 -1.941478 -2.188829 -2.258465 -2.253861 -2.258465 -2.218364
2 -2.261589 -2.008679 -2.240564 -2.261589 -2.209616 -2.255386 -2.261589
3 -2.272472 -2.145274 -2.274500 -2.273162 -2.270017 -2.274500 -2.217785
4 -2.304977 -2.122447 -2.266487 -2.314419 -2.314306 -2.314419 -2.193219

        d35       d36       d37       d38       d39       d40       d41  \
0 -2.264705 -2.264705 -1.754726 -1.783339 -2.264705 -2.082228 -2.264705
1 -2.228129 -2.115470 -1.559824 -2.243014 -2.258465 -2.244237 -2.200919
2 -2.190131 -2.261589 -1.772741 -2.260745 -2.261589 -2.261589 -2.107580
3 -2.193483 -2.274280 -1.686604 -2.179692 -2.190404 -2.265579 -2.001877
4 -2.070420 -2.281815 -1.593093 -2.226729 -2.039310 -2.236987 -2.115215

        d42       d43       d44       d45       d46       d47       d48  \
0 -2.213913 -2.112748 -2.109198 -2.233834 -2.264705 -2.125727 -2.195847
1 -2.257175 -1.827178 -2.258465 -2.243918 -2.253125 -2.245176 -2.246383
2 -2.231726 -1.725262 -2.261589 -2.193373 -2.261496 -2.176671 -2.229890
3 -2.200886 -2.050946 -2.273375 -2.274500 -2.254376 -2.236572 -2.145356
4 -2.100678 -1.840977 -2.227977 -2.141480 -2.277810 -2.108121 -2.312686

        d49       d50       d51       d52       d53       d54       d55  \
0 -1.826038 -2.264705 -2.200957 -2.264705 -2.084250 -2.264705 -2.237384
1 -1.785720 -2.229790 -2.258465 -2.239324 -2.156558 -2.258465 -1.918191
2 -1.941359 -2.261148 -2.225837 -2.240035 -2.227781 -2.231733 -2.053693
3 -1.597930 -2.274500 -2.150162 -2.245699 -2.079147 -2.274500 -2.189509
4 -1.825978 -2.236282 -2.314419 -2.299044 -2.197639 -2.314419 -2.139843

        d56       d57       d58       d59       d60       d61       d62  \
0 -2.139413 -2.264705 -2.128470 -2.264705 -2.264705 -2.196853 -2.202290
1 -2.089434 -2.236462 -2.008789 -2.239560 -2.258465 -2.233065 -2.258465
2 -2.040581 -2.218134 -2.055652 -2.261234 -2.261589 -2.215253 -2.261589
3 -2.229059 -2.172284 -2.099620 -2.274500 -2.274500 -2.237105 -2.274500
4 -2.131050 -2.314419 -2.041263 -2.314376 -2.314419 -2.314419 -2.314419

        d63       d64       d65       d66       d67       d68       d69  \
0 -2.264705 -2.264705 -2.264705 -1.861833 -2.264705 -2.121115 -2.143298
1 -2.258413 -2.257272 -2.258465 -2.257781 -2.249458 -2.257728 -2.250440
2 -2.261589 -2.261589 -2.261589 -2.182716 -2.261589 -2.230970 -2.193001
3 -2.274500 -2.274500 -2.274500 -2.007323 -2.274478 -2.185479 -2.200324
4 -2.313783 -2.314419 -2.314419 -2.101596 -2.279652 -2.314419 -2.192102
```

```
        d70       d71       d72       d73       d74       d75       d76  \
0 -2.264705 -2.176683 -2.264705 -2.264705 -2.132266 -2.205188 -2.245532
1 -2.258465 -2.207799 -2.255582 -2.240722 -2.152171 -2.131113 -2.258465
2 -2.261589 -2.231733 -2.261589 -2.258034 -2.131923 -2.174408 -2.261589
3 -2.274500 -2.274500 -2.274500 -2.268903 -2.084110 -2.222495 -2.273375
4 -2.314419 -2.249895 -2.313092 -2.306605 -2.000283 -2.205904 -2.192019

        d77       d78       d79       d80       d81       d82       d83  \
0 -2.264705 -2.264705 -2.167240 -2.198895 -2.135859 -2.264705 -2.068705
1 -2.229875 -2.237307 -2.131750 -2.159661 -2.250899 -2.258465 -2.217653
2 -2.227776 -2.214393 -1.825558 -2.138177 -2.220624 -2.225439 -2.124128
3 -2.271481 -2.199067 -2.237152 -2.242819 -2.274500 -2.274500 -2.135486
4 -2.263791 -2.298956 -2.314419 -2.170839 -2.237542 -2.303403 -2.139533

        d84       d85       d86       d87       d88       d89       d90  \
0 -2.131226 -2.256353 -2.264705 -2.264705 -2.196079 -2.171596 -2.233834
1 -2.154459 -2.224294 -2.258465 -2.258328 -2.181365 -2.249858 -2.258465
2 -2.213561 -2.260887 -2.261589 -2.261589 -2.261589 -2.093120 -2.261589
3 -2.186383 -2.232372 -2.274500 -2.274500 -2.274500 -2.258919 -2.274500
4 -2.232821 -2.286158 -2.314419 -2.251312 -2.132339 -2.311768 -2.314419

        d91       d92       d93       d94       d95       d96       d97  \
0 -1.954964 -2.195759 -1.803959 -2.207708 -2.264705 -2.252089 -2.264705
1 -1.959073 -2.256241 -1.616490 -2.241060 -2.240085 -2.203317 -2.257887
2 -1.956159 -2.261589 -1.816189 -1.848733 -2.229241 -2.109484 -2.261589
3 -1.885657 -2.026212 -1.769919 -2.140205 -2.273375 -2.213978 -2.272252
4 -1.980276 -2.311195 -1.644558 -2.249169 -2.314419 -2.280771 -2.310449

        d98       d99      d100      d101      d102      d103      d104  \
0 -2.151314 -2.264705 -2.264705 -2.237384 -2.176494 -1.936729 -2.243256
1 -2.223211 -2.246471 -2.258465 -2.252147 -2.230696 -2.256742 -2.256604
2 -2.231733 -2.261516 -2.258710 -2.257149 -2.141864 -2.155560 -2.260251
3 -2.274500 -2.274500 -2.268651 -2.211182 -2.210436 -1.888747 -2.269695
4 -2.314419 -2.314098 -2.314419 -2.233385 -2.215740 -1.888180 -2.299381

       d105      d106      d107      d108      d109      d110      d111  \
0 -2.091369 -2.264705 -2.233834 -2.166199 -2.264705 -2.225888 -2.246301
1 -2.236345 -2.253861 -2.246059 -1.914188 -2.253983 -2.257887 -2.247609
2 -2.181266 -2.259808 -2.214393 -1.850577 -2.258163 -2.133702 -2.193802
3 -2.244338 -2.273375 -2.274500 -2.040770 -2.274500 -2.273562 -2.232762
4 -2.273855 -2.302615 -2.313620 -1.911537 -2.192034 -2.313092 -2.242981

       d112      d113      d114      d115      d116      d117      d118  \
0 -2.264705 -2.264705 -2.180689 -1.876234 -2.264705 -2.264705 -2.264705
1 -2.237307 -2.258465 -2.253142 -2.258465 -2.258465 -2.258465 -2.257887
2 -2.261589 -2.253630 -2.211724 -2.261589 -2.249271 -2.171074 -2.259808
3 -2.274500 -2.274500 -2.217876 -2.046585 -2.274500 -2.274500 -2.274500
```

```
4 -2.314419 -2.314419 -2.280255 -2.268819 -2.314419 -2.314419 -2.314419

       d119      d120      d121      d122      d123      d124      d125  \
0 -2.264705 -2.264705 -2.226632 -2.201685 -2.264705 -2.264705 -2.264705
1 -2.257887 -2.229067 -1.990118 -1.891000 -2.258465 -2.254062 -2.256620
2 -2.258034 -2.261589 -2.149057 -2.139623 -2.261589 -2.261589 -2.259407
3 -2.274500 -2.274500 -2.117879 -2.012696 -2.274500 -2.273879 -2.274500
4 -2.314419 -2.258787 -2.221026 -2.045883 -2.314419 -2.310469 -2.274923

       d126      d127      d128      d129      d130      d131      d132  \
0 -2.196588 -2.264705 -1.962187 -2.176683 -2.135859 -2.264705 -1.985766
1 -2.227477 -2.257470 -1.973554 -2.247022 -2.162211 -2.246937 -2.156657
2 -2.257996 -2.261442 -2.017832 -2.261589 -2.134893 -2.095494 -2.113340
3 -2.071928 -2.229089 -2.020100 -2.274500 -2.236560 -2.238289 -1.954706
4 -2.006501 -2.162895 -2.041490 -2.314419 -2.166495 -2.244244 -2.066932

       d133      d134      d135      d136      d137      d138      d139  \
0 -2.264705 -2.056132 -2.264705 -2.264705 -2.200957 -2.264705 -2.129156
1 -2.246212 -2.242829 -2.258465 -2.187948 -2.195820 -2.258465 -1.939228
2 -2.261589 -2.231733 -2.261589 -2.257044 -2.225636 -2.261589 -1.827653
3 -2.274500 -1.993371 -2.274500 -2.274498 -2.270017 -2.274500 -2.139894
4 -2.312790 -2.107652 -2.314419 -2.164207 -2.151262 -2.314419 -1.894829

       d140      d141      d142      d143      d144      d145      d146  \
0 -2.255405 -2.092796 -2.264705 -2.233834 -2.176576 -2.255405 -2.264705
1 -2.250440 -2.227043 -2.156036 -2.258465 -2.244187 -2.256733 -2.258193
2 -2.251881 -2.104072 -2.110598 -2.261589 -2.150369 -2.259808 -2.258162
3 -2.225174 -2.232459 -2.274246 -2.274500 -2.241688 -2.246039 -2.274500
4 -2.242141 -2.264048 -2.311370 -2.313755 -2.254888 -2.283627 -2.314419

       d147      d148      d149         0    1         2         3         4  \
0 -2.264705 -2.241351 -2.225888  0.000000  0.0  0.000000  0.000000  0.000000
1 -2.255866 -2.257384 -2.235050  0.033177  0.0  0.010711  0.001567  0.000784
2 -2.261589 -2.137481 -2.230928  0.080240  0.0  0.000000  0.003593  0.000000
3 -2.271636 -2.230984 -2.274500  0.000000  0.0  0.000000  0.005141  0.000000
4 -2.312400 -2.314419 -2.282669  0.062099  0.0  0.000000  0.000000  0.004283

          5         6         7         8         9        10        11   12  \
0  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.0
1  0.008359  0.031348  0.013845  0.000000  0.000784  0.003135  0.007576  0.0
2  0.000000  0.025150  0.021557  0.000000  0.007186  0.007186  0.000000  0.0
3  0.010283  0.005141  0.000000  0.033419  0.000000  0.000000  0.000000  0.0
4  0.012848  0.031406  0.008565  0.000000  0.000000  0.019272  0.000000  0.0

         13        14        15        16        17        18        19  \
0  0.000000  0.000000  0.064103  0.000000  0.000000  0.000000  0.000000
1  0.003135  0.006008  0.019070  0.038140  0.002351  0.007053  0.012278
```

```
2  0.014371  0.028743  0.035928  0.007186  0.000000  0.000000  0.000000
3  0.000000  0.000000  0.015424  0.002571  0.041131  0.000000  0.000000
4  0.000000  0.000000  0.042113  0.014989  0.010707  0.000000  0.000000

         20        21        22        23   24        25        26   27  \
0  0.038462  0.000000  0.051282  0.000000  0.0  0.000000  0.000000  0.0
1  0.003918  0.000784  0.006008  0.004702  0.0  0.001567  0.024033  0.0
2  0.000000  0.000000  0.014371  0.028743  0.0  0.000000  0.000000  0.0
3  0.010283  0.000000  0.012853  0.000000  0.0  0.000000  0.000000  0.0
4  0.000000  0.010707  0.000000  0.000000  0.0  0.039971  0.027837  0.0

         28        29        30        31        32        33        34  \
0  0.000000  0.000000  0.025641  0.000000  0.000000  0.000000  0.000000
1  0.007576  0.000784  0.009143  0.000000  0.008621  0.013062  0.072362
2  0.032335  0.000000  0.007186  0.017964  0.089820  0.010778  0.000000
3  0.000000  0.000000  0.020566  0.002571  0.000000  0.000000  0.000000
4  0.017131  0.000000  0.004283  0.089222  0.018558  0.019272  0.000000

         35   36        37        38        39        40        41        42  \
0  0.000000  0.0  0.000000  0.012821  0.000000  0.000000  0.000000  0.000000
1  0.000784  0.0  0.005486  0.000000  0.002351  0.022205  0.028213  0.043887
2  0.003593  0.0  0.003593  0.000000  0.021557  0.014371  0.000000  0.032335
3  0.020566  0.0  0.002571  0.007712  0.000000  0.017995  0.000000  0.000000
4  0.000000  0.0  0.000000  0.000000  0.012848  0.000000  0.008565  0.031406

         43        44   45   46        47        48        49        50  \
0  0.141026  0.038462  0.0  0.0  0.000000  0.000000  0.000000  0.000000
1  0.000000  0.000784  0.0  0.0  0.013062  0.078631  0.001567  0.003918
2  0.000000  0.000000  0.0  0.0  0.067066  0.046707  0.000000  0.102994
3  0.000000  0.007712  0.0  0.0  0.000000  0.000000  0.000000  0.019709
4  0.000000  0.000000  0.0  0.0  0.008565  0.039971  0.000000  0.021413

         51        52        53        54        55   56        57        58  \
0  0.000000  0.000000  0.000000  0.000000  0.000000  0.0  0.000000  0.012821
1  0.017503  0.000000  0.000000  0.009927  0.003135  0.0  0.000000  0.000000
2  0.000000  0.000000  0.000000  0.000000  0.000000  0.0  0.000000  0.000000
3  0.000000  0.041131  0.005141  0.000000  0.000000  0.0  0.005141  0.015424
4  0.000000  0.000000  0.002141  0.000000  0.086367  0.0  0.000000  0.000000

        59        60        61        62        63   64        65        66  \
0  0.00000  0.000000  0.051282  0.025641  0.025641  0.0  0.000000  0.000000
1  0.03605  0.000000  0.001567  0.000000  0.013323  0.0  0.000000  0.000000
2  0.00000  0.000000  0.000000  0.000000  0.000000  0.0  0.000000  0.000000
3  0.00000  0.002571  0.033419  0.025707  0.000000  0.0  0.277635  0.161954
4  0.00000  0.000000  0.000000  0.000000  0.019272  0.0  0.000000  0.000000

    67        68   69        70        71        72        73   74        75  \
```

```
0  0.0  0.000000  0.0  0.000000  0.000000  0.000000  0.000000  0.0  0.000000
1  0.0  0.007837  0.0  0.006008  0.007837  0.003135  0.016196  0.0  0.001567
2  0.0  0.000000  0.0  0.000000  0.014371  0.000000  0.010778  0.0  0.000000
3  0.0  0.000000  0.0  0.000000  0.000000  0.000000  0.007712  0.0  0.000000
4  0.0  0.000000  0.0  0.004283  0.017131  0.012848  0.050678  0.0  0.000000

         76        77        78   79   80        81   82        83        84  \
0  0.000000  0.000000  0.273504  0.0  0.0  0.000000  0.0  0.000000  0.000000
1  0.006270  0.003135  0.000000  0.0  0.0  0.000000  0.0  0.016196  0.063480
2  0.023952  0.003593  0.000000  0.0  0.0  0.000000  0.0  0.000000  0.000000
3  0.005141  0.000000  0.000000  0.0  0.0  0.042845  0.0  0.000000  0.032562
4  0.014989  0.014989  0.000000  0.0  0.0  0.000000  0.0  0.012848  0.014989

         85        86        87        88        89        90        91  \
0  0.000000  0.000000  0.089744  0.000000  0.085470  0.000000  0.000000
1  0.001567  0.000000  0.000000  0.002351  0.010711  0.014629  0.062696
2  0.007186  0.000000  0.000000  0.000000  0.003593  0.068263  0.067066
3  0.000000  0.033419  0.041131  0.000000  0.015424  0.010283  0.000000
4  0.000000  0.000000  0.000000  0.000000  0.002141  0.034261  0.049251

         92        93        94        95        96   97   98        99
0  0.000000  0.064103  0.000000  0.000000  0.000000  0.0  0.0  0.000000
1  0.014890  0.014368  0.025340  0.017241  0.049112  0.0  0.0  0.021160
2  0.000000  0.076647  0.000000  0.000000  0.000000  0.0  0.0  0.000000
3  0.002571  0.000000  0.000000  0.000000  0.000000  0.0  0.0  0.005141
4  0.000000  0.010707  0.002141  0.000000  0.075660  0.0  0.0  0.019272
```

[29]:
```
# booking is all that matter since everything else is a non-starter filter␣
 ↪everything but bookings

train = train.loc[train['is_booking'] == 1]
```

[30]:
```
# what are we left with
train.shape
```

[30]:
```
(20032, 276)
```

[39]:
```
#Make a copy before dropping elements
train_copy2 = train.copy(deep=True)


# use if needed
# copy the data in case need to go back
#train = train_copy1.copy(deep=True)
```

[32]:
```
# make split between features and target

```

```
X = train.drop(['user_id', 'hotel_cluster', 'is_booking'], axis=1)
y = train.hotel_cluster
```

[42]:
```
#Split the data in training and Test
X_train, X_test, y_train, y_test = train_test_split (X, y, test_size=0.2,␣
 ↪random_state=0)
```

[47]:
```
X_train.shape
```

[47]: (16025, 273)

[48]:
```
X_test.shape
```

[48]: (4007, 273)

[43]:
```
# bring in Gaussian naiv bayes as first algorithm test

from sklearn.naive_bayes import GaussianNB


gnb = make_pipeline(preprocessing.StandardScaler(),
                    GaussianNB(priors=None))

# get accuracy
np.mean(cross_val_score(gnb, X_train, y_train, cv=10))
```

[43]: 0.10352557587482272

[44]:
```
np.mean(cross_val_score(gnb, X_test, y_test, cv=10))
```

```
C:\Users\Tushar\anaconda3\lib\site-
packages\sklearn\model_selection\_split.py:666: UserWarning: The least populated
class in y has only 3 members, which is less than n_splits=10.
  warnings.warn(("The least populated class in y has only %d"
```

[44]: 0.1005785536159601

[34]:
```
# run KNN as second test

from sklearn.neighbors import KNeighborsClassifier

knn = make_pipeline(preprocessing.StandardScaler(),
                    KNeighborsClassifier(n_neighbors=5))
np.mean(cross_val_score(knn, X_train, y_train, cv=10, scoring='accuracy'))
```

[34]: 0.25643954228338134

[49]:
```
# run KNN as second test

from sklearn.neighbors import KNeighborsClassifier

knn = make_pipeline(preprocessing.StandardScaler(),
                    KNeighborsClassifier(n_neighbors=5))
np.mean(cross_val_score(knn, X_test, y_test, cv=10, scoring='accuracy'))
```

```
C:\Users\Tushar\anaconda3\lib\site-
packages\sklearn\model_selection\_split.py:666: UserWarning: The least populated
class in y has only 3 members, which is less than n_splits=10.
  warnings.warn(("The least populated class in y has only %d"
```

[49]:
```
0.2580417705735661
```

[51]:
```
# run Random Forest

rfc = make_pipeline(preprocessing.StandardScaler(),
                    ␣
 ↪RandomForestClassifier(n_estimators=273,max_depth=10,random_state=0))

np.mean(cross_val_score(rfc, X_train, y_train, cv=10))
```

[51]:
```
0.24817469273825682
```

[52]:
```
np.mean(cross_val_score(rfc, X_test, y_test, cv=10))
```

```
C:\Users\Tushar\anaconda3\lib\site-
packages\sklearn\model_selection\_split.py:666: UserWarning: The least populated
class in y has only 3 members, which is less than n_splits=10.
  warnings.warn(("The least populated class in y has only %d"
```

[52]:
```
0.24656483790523692
```

[35]:
```
# run svm

from sklearn import svm

svm = make_pipeline(preprocessing.StandardScaler(), svm.
 ↪SVC(decision_function_shape='ovo'))
np.mean(cross_val_score(svm, X, y, cv=10))
```

[35]:
```
0.324280146646298
```

2 My original model

[55]:
```
otrain = train_copy1.copy(deep=True)
```

[56]:
```
otrain.shape
```

[56]:
```
(241179, 24)
```

[57]:
```
otrain.columns
```

[57]:
```
Index(['date_time', 'site_name', 'posa_continent', 'user_location_country',
       'user_location_region', 'user_location_city',
       'orig_destination_distance', 'user_id', 'is_mobile', 'is_package',
       'channel', 'srch_ci', 'srch_co', 'srch_adults_cnt', 'srch_children_cnt',
       'srch_rm_cnt', 'srch_destination_id', 'srch_destination_type_id',
       'is_booking', 'cnt', 'hotel_continent', 'hotel_country', 'hotel_market',
       'hotel_cluster'],
      dtype='object')
```

[58]:
```
otrain['date_time_year'] = pd.Series(otrain.date_time, index = otrain.index)
otrain['date_time_month'] = pd.Series(otrain.date_time, index = otrain.index)
```

[59]:
```
# conver year and month from datetime

from datetime import datetime # import date time to get year and month

otrain.date_time_year = otrain.date_time_year.apply(lambda x: get_year(x)) #␣
 ↪for year


otrain.date_time_month = otrain.date_time_month.apply(lambda x: get_month(x)) #␣
 ↪for month

del otrain['date_time'] #delete the original as it is not needed
```

[60]:
```
# get year and month for check in date

otrain['srch_ci_year'] = pd.Series(otrain.srch_ci, index=otrain.index)
otrain['srch_ci_month'] = pd.Series(otrain.srch_ci, index=otrain.index)

# convert year & months to int

otrain.srch_ci_year = otrain.srch_ci_year.apply(lambda x: get_year(x))
otrain.srch_ci_month = otrain.srch_ci_month.apply(lambda x: get_month(x))

# remove the srch_ci column
del otrain['srch_ci']
```

[61]:
```
# get year and month for check out date

otrain['srch_co_year'] = pd.Series(otrain.srch_co, index=otrain.index)
otrain['srch_co_month'] = pd.Series(otrain.srch_co, index=otrain.index)

# convert year & months to int

otrain.srch_co_year = otrain.srch_co_year.apply(lambda x: get_year(x))
otrain.srch_co_month = otrain.srch_co_month.apply(lambda x: get_month(x))

# remove the srch_co column

del otrain['srch_co']
```

[62]:
```
otrain.head()
```

[62]:

	site_name	posa_continent	user_location_country	user_location_region
0	2	3	66	174
1	2	3	66	311
2	2	3	66	294
3	2	3	66	332
4	2	3	66	314

	user_location_city	orig_destination_distance	user_id	is_mobile
0	24103	2323.5232	802499	0
1	25538	2288.6121	85229	0
2	40046	587.6970	755217	0
3	55121	2234.4394	160733	0
4	47869	839.0087	1078493	0

	is_package	channel	srch_adults_cnt	srch_children_cnt	srch_rm_cnt
0	1	9	2	0	1
1	0	9	3	1	1
2	1	9	2	0	1
3	1	9	2	0	1
4	0	9	4	0	1

	srch_destination_id	srch_destination_type_id	is_booking	cnt
0	1442	3	0	1
1	8272	1	0	1
2	11321	1	0	1
3	1152	1	1	1
4	8284	1	0	4

	hotel_continent	hotel_country	hotel_market	hotel_cluster
0	4	125	177	44
1	2	50	659	59

```
2                2             50           642             22
3                4             47          1502             65
4                2             50           685              6

   date_time_year  date_time_month  srch_ci_year  srch_ci_month  srch_co_year  \
0            2014                5          2014              7          2014
1            2013                6          2013              7          2013
2            2014               10          2014             12          2014
3            2014                8          2015              1          2015
4            2014                3          2014              4          2014

   srch_co_month
0              7
1              7
2             12
3              1
4              4
```

[63]:
```
otrain.columns
```

[63]:
```
Index(['site_name', 'posa_continent', 'user_location_country',
       'user_location_region', 'user_location_city',
       'orig_destination_distance', 'user_id', 'is_mobile', 'is_package',
       'channel', 'srch_adults_cnt', 'srch_children_cnt', 'srch_rm_cnt',
       'srch_destination_id', 'srch_destination_type_id', 'is_booking', 'cnt',
       'hotel_continent', 'hotel_country', 'hotel_market', 'hotel_cluster',
       'date_time_year', 'date_time_month', 'srch_ci_year', 'srch_ci_month',
       'srch_co_year', 'srch_co_month'],
      dtype='object')
```

[64]:
```
# make split between features and target

Xo = otrain.drop(['site_name','user_location_region','user_id', 'is_mobile',␣
 ↪'srch_adults_cnt',
                  'srch_children_cnt','srch_destination_id',␣
 ↪'srch_destination_type_id','hotel_continent',
                  'hotel_cluster',], axis=1)
yo = otrain.hotel_cluster
```

[65]:
```
# split otrain into X and y with only variables that are needed

Xo_train, Xo_test, yo_train, yo_test = train_test_split (Xo, yo, test_size=0.2,␣
 ↪random_state=0)
```

[66]:
```
knn = make_pipeline(preprocessing.StandardScaler(),
                    KNeighborsClassifier(n_neighbors=5))
np.mean(cross_val_score(knn, Xo_train, yo_train, cv=10, scoring='accuracy'))
```

[66]: 0.037311534858665 87

[ ]: