ASSIGNMENT TASK 2
WEB LOG DATA ANALYTICS

SIT742 – MODERN DATA SCIENCE

Unit Chair AsPr Gang Li

OLUWATADE JOB ADEKOLA
Student ID: 215383256
SAOWALUK VONGWISEDSORAWUT
Student ID: 215013027

Contents

Web Log Data Analytics

Introduction

After doing the initial analysis and discovering relationships about the TULIP data set, it is imperative that as a data analyst contracted to solve problems and generate insights with this large and unstructured dataset to proceed and do some predictive analysis on the data.

In this second task, we will work on discovering more analytics insights from this dataset by being specific with the analytic process. Therefore, what will be done is to create a session ID by combining some variables from the table generated from the web log data. These combined variables will form a unique identification (ID) to identify user activity pattern in the Hotel Tulip website.

To make this work, the URI stem, which shows the page link accessed can be used to group the SQL query. In addition, from the URI path, some filter can be done to eliminate stems with (404, js, jquery etc.)

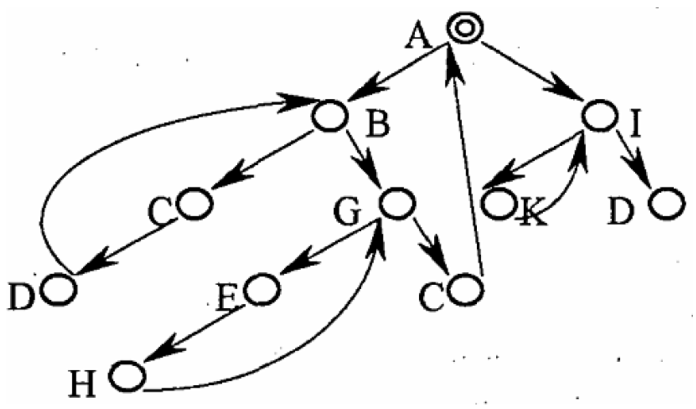
Also, as the new session ID column is created, the dimensionality of the dataset is reduced as well because several columns are combined to form one.

After the session ID has been created, we can combine all the columns that would be used for analysis with the newly generated session ID column. World map plots can be done to identify where the most sessions are created by grouping SQL query by User IP column. Also, more analysis like identifying the through the user agent a particular kind of user's activity or more so, through the session ID, we can Identify user activity and their activity pattern on the Tulip website.

Furthermore, by playing with time, queries like getting insights on a set day where there seems to be possible traffic on the site. For instance during summer where more people tend to travel or during festive periods. More insights can be generated during these times to understand user needs or even probably prescribe what customers might like during these times.

Problem Description

User access pattern, which is a method with its concept based on user access path, which is an important sequential pattern in web usage mining. After pre-processing Hotel Tulip log data, we can get user access paths from user session files. The figure below shows a sample user's browsing path through which User session id accesses certain web site: A-B-C-D-B-G-E-H-G-C-A-I-K [2].



Web Log Data Preparation and Exploration

Firstly, minimize the dataset by selecting only the variables we interested and related to the analysis purpose. Step by step procedures on how the tasks described above are implemented are shown in the screen shots below:

The table shows how the new session ID column was formed.

```
root
 |-- cs_User_Agent: string (nullable = true)
 |-- cs_ip: string (nullable = true)
 |-- cs_uri_stem: string (nullable = true)
 |-- date: string (nullable = true)
 |-- date_time: timestamp (nullable = true)
 |-- path: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- sc_status: long (nullable = true)
 |-- sc_substatus: long (nullable = true)
 |-- sc_win32_status: long (nullable = true)
 |-- session_id: string (nullable = true)
 |-- time: string (nullable = true)
 |-- time_taken: long (nullable = true)
```

The new Session ID column which combines date, user IP and User Agent.

Date — IP address — User_agent

session_id
2014-08-01202.140.108.99Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_1_2+like+Mac+OS+X)+AppleWebKit/537.51.2+(KHTML,+like+Gecko)+Version/7.0+Mc
2014-08-01207.6.118.176Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_1_2+like+Mac+OS+X)+AppleWebKit/537.51.2+(KHTML,+like+Gecko)+Mobile/11D257
2014-08-01207.6.118.176Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_1_2+like+Mac+OS+X)+AppleWebKit/537.51.2+(KHTML,+like+Gecko)+Mobile/11D257
2014-08-01207.6.118.176Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_1_2+like+Mac+OS+X)+AppleWebKit/537.51.2+(KHTML,+like+Gecko)+Mobile/11D257
2014-08-01207.6.118.176Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_1_2+like+Mac+OS+X)+AppleWebKit/537.51.2+(KHTML,+like+Gecko)+Mobile/11D257
2014-08-01207.6.118.176Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_1_2+like+Mac+OS+X)+AppleWebKit/537.51.2+(KHTML,+like+Gecko)+Mobile/11D257
2014-08-01207.6.118.176Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_1_2+like+Mac+OS+X)+AppleWebKit/537.51.2+(KHTML,+like+Gecko)+Mobile/11D257
2014-08-01207.6.118.176Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_1_2+like+Mac+OS+X)+AppleWebKit/537.51.2+(KHTML,+like+Gecko)+Mobile/11D257
2014-08-01207.6.118.176Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_1_2+like+Mac+OS+X)+AppleWebKit/537.51.2+(KHTML,+like+Gecko)+Mobile/11D257
2014-08-01207.6.118.176Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_1_2+like+Mac+OS+X)+AppleWebKit/537.51.2+(KHTML,+like+Gecko)+Mobile/11D257

The session ID number generated using hash function into unique number.

session_id	date	time	date_time	cs_ip
-1811938738	2014-08-01	00:00:25	2014-08-01 00:00:25.0	202.140.108.99
-1442062705	2014-08-01	00:00:28	2014-08-01 00:00:28.0	207.6.118.176
-1738057240	2014-08-01	00:00:40	2014-08-01 00:00:40.0	14.199.63.188
-1442062705	2014-08-01	00:00:48	2014-08-01 00:00:48.0	207.6.118.176
-1738057240	2014-08-01	00:00:48	2014-08-01 00:00:48.0	14.199.63.188
1674396148	2014-08-01	00:01:12	2014-08-01 00:01:12.0	61.92.230.63
1561145465	2014-08-01	00:01:30	2014-08-01 00:01:30.0	119.236.43.88
1561145465	2014-08-01	00:01:31	2014-08-01 00:01:31.0	119.236.43.88
133623124	2014-08-01	00:01:41	2014-08-01 00:01:41.0	61.92.230.63
1674396148	2014-08-01	00:01:44	2014-08-01 00:01:44.0	61.92.230.63
89641610	2014-08-01	00:01:47	2014-08-01 00:01:47.0	203.198.136.211

Created bu hash funcion

At this stage, the variables shown are the finally selected variables to be used for analysis but the web page column contains 404, which is a server code for returning an empty page. This is not needed for our analysis. So, the next step is to write an SQL query to exclude the 404 code from the results returned.

session_id	date	time	date_time	cs_ip	page	time_taken	sc_status
-1811938738	2014-08-01	00:00:25	2014-08-01 00:00:25.0	202.140.108.99	LocationContacts.aspx	19	200
-1442062705	2014-08-01	00:00:28	2014-08-01 00:00:28.0	207.6.118.176	home.aspx	210	302
-1738057240	2014-08-01	00:00:40	2014-08-01 00:00:40.0	14.199.63.188	default.aspx	20	200
-1442062705	2014-08-01	00:00:48	2014-08-01 00:00:48.0	207.6.118.176	GuestRooms.aspx	327	200
-1738057240	2014-08-01	00:00:48	2014-08-01 00:00:48.0	14.199.63.188	default.aspx	20	200
1674396148	2014-08-01	00:01:12	2014-08-01 00:01:12.0	61.92.230.63	404.aspx	27	404
1561145465	2014-08-01	00:01:30	2014-08-01 00:01:30.0	119.236.43.88	Dining.aspx	35	200
1561145465	2014-08-01	00:01:31	2014-08-01 00:01:31.0	119.236.43.88	404.aspx	27	404
133623124	2014-08-01	00:01:41	2014-08-01 00:01:41.0	61.92.230.63	LocationContacts.aspx	53	200
1674396148	2014-08-01	00:01:44	2014-08-01 00:01:44.0	61.92.230.63	404.aspx	26	404
89641610	2014-08-01	00:01:47	2014-08-01 00:01:47.0	203.198.136.211	offers.aspx	129	200
89641610	2014-08-01	00:01:52	2014-08-01 00:01:52.0	203.198.136.211	404.aspx	22	200
1674396148	2014-08-01	00:02:03	2014-08-01 00:02:03.0	61.92.230.63	404.aspx	29	404
944277774	2014-08-01	00:02:04	2014-08-01 00:02:04.0	202.140.108.93	Dining.aspx	25	200

The screenshot below shows that the 404 server code has been eliminated by show a query result for all the 404 pages requested.

session_id	date	time	date_time	cs_ip	web_page	time_taken	sc_status
1674396148	2014-08-01	00:01:12	2014-08-01 00:01:12.0	61.92.230.63	404.aspx	27	404
1561145465	2014-08-01	00:01:31	2014-08-01 00:01:31.0	119.236.43.88	404.aspx	27	404
1674396148	2014-08-01	00:01:44	2014-08-01 00:01:44.0	61.92.230.63	404.aspx	26	404
89641610	2014-08-01	00:01:52	2014-08-01 00:01:52.0	203.198.136.211	404.aspx	22	200
1674396148	2014-08-01	00:02:03	2014-08-01 00:02:03.0	61.92.230.63	404.aspx	29	404
1674396148	2014-08-01	00:02:17	2014-08-01 00:02:17.0	61.92.230.63	404.aspx	27	404
415595108	2014-08-01	00:02:32	2014-08-01 00:02:32.0	121.229.123.168	404.aspx	241	200
1674396148	2014-08-01	00:02:32	2014-08-01 00:02:32.0	61.92.230.63	404.aspx	31	404

Before we can plot on map, we have to get the country code which is cca3 the cca3 can be identify from the IP Address of user. Moreover, the country.csv file which we have download from the web country code [1]. The table below shows the IP Address, code country (cca2 and cca3) and country name.

ip	cca2	cca3	cn
185.25.49.181	LT	LTU	Lithuania
80.220.130.84	FI	FIN	Finland
81.197.28.112	FI	FIN	Finland
109.86.75.187	UA	UKR	Ukraine
193.201.224.18	UA	UKR	Ukraine
195.211.155.253	UA	UKR	Ukraine
195.216.206.114	UA	UKR	Ukraine
46.119.115.111	UA	UKR	Ukraine
91.250.15.69	UA	UKR	Ukraine

Using map to visualise the countries areas with high frequency of page error (404.aspx), from map plot below we can see that USA, Canada and China have a high number of error page.

After eliminating the error code from the URL, using the Session ID, a unique session ID is selected using the WHERE SQL clause to identify the pattern of access by the particular user. The table below shows the date, time and URL pattern access of a particular user below.

session_id	date_time	cs_ip	page	time_taken
-1442062705	2014-08-01 00:00:28.0	207.6.118.176	home.aspx	210
-1442062705	2014-08-01 00:00:28.0	207.6.118.176	null	442
-1442062705	2014-08-01 00:00:29.0	207.6.118.176	mobile.css	443
-1442062705	2014-08-01 00:00:29.0	207.6.118.176	mobile.js	220
-1442062705	2014-08-01 00:00:30.0	207.6.118.176	image_dining.ashx	293
-1442062705	2014-08-01 00:00:30.0	207.6.118.176	btn_sc.gif	305
-1442062705	2014-08-01 00:00:30.0	207.6.118.176	nav_findus.gif	185
-1442062705	2014-08-01 00:00:30.0	207.6.118.176	image_offers.jpg	726
-1442062705	2014-08-01 00:00:30.0	207.6.118.176	image_main.jpg	403

Unique Session_id

From the table above, it's realised that more filter has to be done to the web page column in order to return only the ".aspx" page as the "css" and "js" are not web pages.

After the filter has been done, the table below was generated for the selected user id and it is ordered by time. From the table below, it can be seen that the user after accessing the home page, went on to access the "Guestrooms" page and then back to the homepage. If further query is run on the whole dataset, we can deduce even much more insights can be generated.

session_id	date	time	date_time	cs_ip	page	time_taken
-1442062705	2014-08-01	00:00:28	2014-08-01 00:00:28.0	207.6.118.176	home.aspx	210
-1442062705	2014-08-01	00:00:48	2014-08-01 00:00:48.0	207.6.118.176	GuestRooms.aspx	327
-1442062705	2014-08-01	00:21:50	2014-08-01 00:21:50.0	207.6.118.176	home.aspx	300

No error page

Having generated these insights, we can do even more analysis by doing a trace back procedure where we can focus on the user agent. We can try to understand the devices that are frequently used or the kind of browsers that are best used. This can be achieved by selecting the string session ID column and filtering the 404 pages, .com pages that contains bot and other pages that are not like .aspx pattern. The table below shows the outcome of this analysis.

```
1 sqlContext.sql("select session_id, page from SQLTB_1  where page like '%.aspx%' ").show(20,False)
```

session_id	page
-1811938738	LocationContacts.aspx
-1442062705	home.aspx
-1738057240	default.aspx
-1442062705	GuestRooms.aspx
-1738057240	default.aspx
1674396148	404.aspx
1561145465	Dining.aspx
1561145465	404.aspx
133623124	LocationContacts.aspx

404 Page Error

The table shows the session ID (users) that got redirected to 404 page

session_id	cs_ip	page_count
15787048	10.120.7.25	610
-724338416	10.120.7.25	230
-1186583656	10.120.7.25	204
-1789420754	67.192.185.67	107
378297857	124.217.186.135	58
-2121028001	10.120.7.23	53
-1748703174	203.145.92.73	50

Sequence of page that session_id go through

Table below showing the sequence of the web page that this particular user go through, we can see that from the attribute “time” tells us which web page they went through and we can estimate the spend on each page as well.

session_id	time	page	time_taken	sc_status	Sequence
1872063208	02:48:58	default.aspx	524	302	1
1872063208	02:49:49	rooms.aspx	359	200	2
1872063208	02:54:45	home.aspx	456	200	3
1872063208	02:55:00	offers.aspx	161	200	4
1872063208	05:22:20	location-and-contacts.aspx	92	200	5
1872063208	05:31:18	rooms.aspx	434	200	6
1872063208	06:00:16	rooms.aspx	400	200	7
1872063208	06:00:24	rooms.aspx	306	302	8
1872063208	06:00:24	rooms.aspx	162	200	9
1872063208	06:00:33	rooms.aspx	594	302	10
1872063208	06:00:34	rooms.aspx	430	200	11
1872063208	06:01:06	rooms.aspx	469	302	12
1872063208	06:01:06	rooms.aspx	296	200	13
1872063208	06:02:29	our-city.aspx	639	200	14
1872063208	06:02:59	rooms.aspx	361	200	15
1872063208	06:04:21	rooms.aspx	441	302	16
1872063208	06:04:22	rooms.aspx	521	200	17

Web Log Data Analysis

Implementing FP Tree Pattern

The frequency tree pattern is algorithm which implements a tree structure to identify commonly occurring items in a dataset. In this case, we have implemented FP tree algorithm to show frequently accessed web pages or webpages most often accessed together i.e. association relationship between them. This can be used for further insights. For instance, the clients could opt to use the insight to post ads on the highest frequently viewed page.

```
from pyspark.mllib.fpm import FPGrowth
data = sc.parallelize(webpage)
web_page_visits = data.map(lambda line: line.strip().split(' '))
model = FPGrowth.train(web_page_visits, minSupport=0.02, numPartitions=10)
result = model.freqItemsets().collect()
for fi in result:
    print(fi)

```

From the underlined FP growth statement shown in the FP growth model shown above, the web page visit column from the weblog database is trained to incorporate the FP growth algorithm and then partitions are equally created. Then the algorithm builds a tree-like pattern of representation of the frequencies.

The screenshot below shows the top 2 frequently accessed webpages. It could be easily induced that there are usually lots of offers or probably the hostel is best known for its dining services.

```
FreqItemset(items=[u'dining.aspx'], freq=10767)
FreqItemset(items=[u'offers.aspx'], freq=10352)
```

The picture below shows the Frequent Pattern, we can see that in the yellow box showing the only one web page pattern that have the highest frequent which is "**dining.aspx**" and "**offer.aspx**" is the second web page that has a high frequent.

Additionally, the red box shows the 2 path pattern that has the highest frequent which mean 2406 user visited web page by went through the same pattern ("**room.aspx** > **offer.aspx**"), and the second place is "**about-the-hotel.aspx** > **room.aspx** " having 1650 user using the same pattern.

```
FreqItemset(items=[u'home.aspx'], freq=2096)
FreqItemset(items=[u'home.aspx', u'offers.aspx'], freq=812)
FreqItemset(items=[u'dining.aspx'], freq=10767)
FreqItemset(items=[u'GuestRooms.aspx'], freq=2061)
FreqItemset(items=[u'offers.aspx'], freq=10352)
FreqItemset(items=[u'offers.aspx', u'dining.aspx'], freq=2428)
FreqItemset(items=[u'rooms.aspx'], freq=7107)
FreqItemset(items=[u'rooms.aspx', u'offers.aspx'], freq=2406)
FreqItemset(items=[u'rooms.aspx', u'dining.aspx'], freq=1395)
FreqItemset(items=[u'LocationContacts.aspx'], freq=1791)
FreqItemset(items=[u'Facilities.aspx'], freq=1586)
FreqItemset(items=[u'Dining.aspx'], freq=5973)
FreqItemset(items=[u'our-city.aspx'], freq=1263)
FreqItemset(items=[u'about-the-hotel.aspx'], freq=4606)
FreqItemset(items=[u'about-the-hotel.aspx', u'rooms.aspx'], freq=1650)
FreqItemset(items=[u'about-the-hotel.aspx', u'rooms.aspx', u'offers.aspx'], freq=815)
FreqItemset(items=[u'about-the-hotel.aspx', u'offers.aspx'], freq=1434)
```

Plotting Map for countries that have regular frequent access to the Top Two Webpages.

Plotting for Offers Page

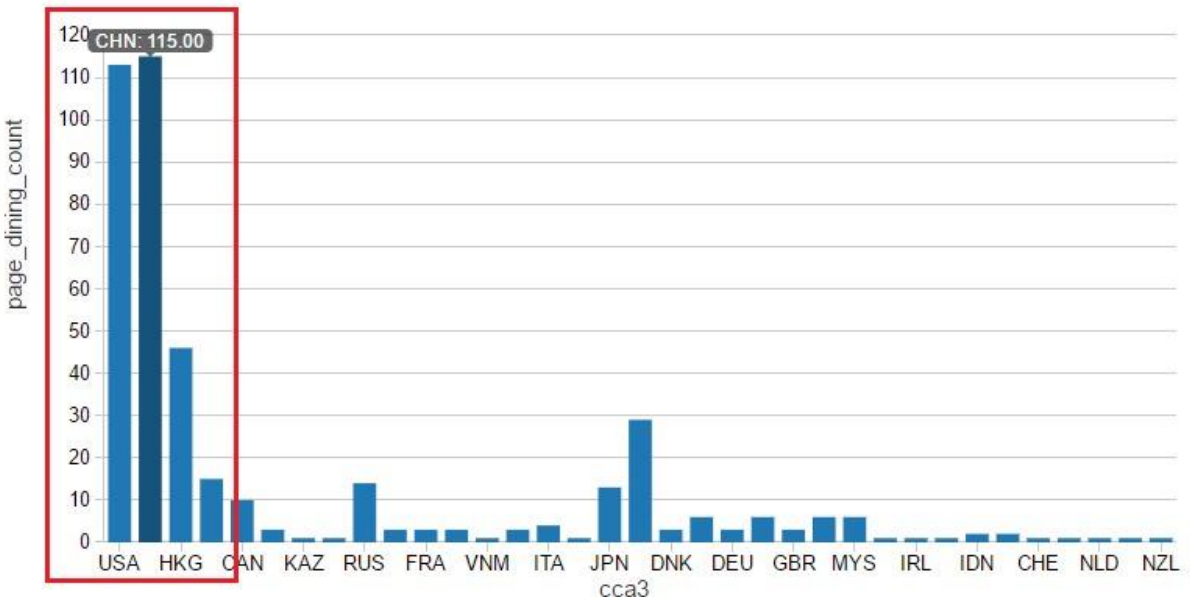
session_id	cs_ip	offer_page_count
-1186583656	10.120.7.25	26
1571920094	121.54.54.148	13
-1434289964	162.250.233.57	12
1276963349	10.120.7.12	12
322910954	157.55.39.14	9
1011933523	119.246.184.40	9
1687913708	114.42.237.253	9
953680556	180.155.43.49	8
550254821	157.55.39.255	7
1729092307	27.109.189.245	7

The table shows a count of the number of times offer page was visited, the session ID and IP address which helps in finding their location.

From the map above, we try to do analysis for top 2 web pages frequently accessed in a five-day period. It was discovered that TULIP users from US and China head the count of most frequent users accessing the offers page. This could further likewise, we'd also do a count for the dining webpage

Plotting for Dining Page

The world map shows that China has the highest number of page request for dining. More understanding about this insight can be discovered in the sense that the tulip team might expand their menu for Chinese food and increase varieties as well.



Another graphical representation of dining page count with china representing the highest population.

Conclusion

We will realize that after so much analysis and machine learning approach was implemented, we can draw some useful conclusion and recommendations.

Firstly, focusing on Session ID, we can draw lots of insights about customer access behavior and experience. Then we can provide recommendations based on this to improve marketing and customer experience.

In addition, FP tree was implemented which an algorithm that is an example of association rule is mining. It works in such a way that it returns webpages that most often accessed together. This could be used to as a business strategy to create recommendation systems that might improve their services as well as understand user demands.

Using user access pattern in conjunction with the FP tree model would far benefit Tulip team about understanding individual user usage pattern and sometimes help in providing individualized recommendations.

References

[1] Gist. (2017). *wikipedia-iso-country-codes.csv*. [online] Available at: https://gist.github.com/radcliff/f09c0f88344a7fcef373 [Accessed 28 May 2017]

[2] Wang, X., Ouyang, Y., Hu, X. and Zhang, Y., 2004, May. Discovery of user frequent access patterns on Web usage mining. In *Computer Supported Cooperative Work in Design, 2004. Proceedings. The 8th International Conference on* (Vol. 1, pp. 765-769). IEEE.